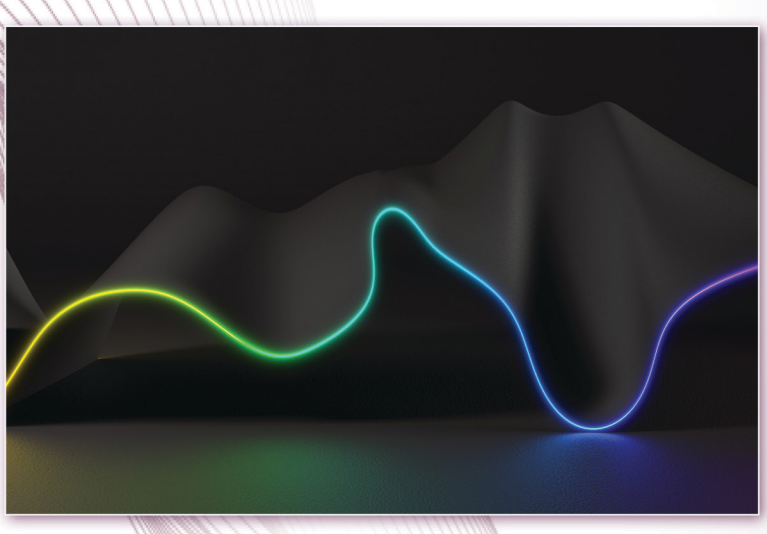
ELEMENTS OF CLASSICAL AND GEOMETRIC OPTIMIZATION



DEBASISH ROY
G. VISWESWARA RAO

CRC Press
Taylor & Francis Group

Elements of Classical and Geometric Optimization

This comprehensive textbook covers both classical and geometric aspects of optimization using methods, deterministic and stochastic, in a single volume and in a language accessible to non-mathematicians. It will serve as an ideal study material for senior undergraduate and graduate students in the fields of civil, mechanical, aerospace, electrical, electronics, and communication engineering.

The book includes:

- Derivative-based Methods of Optimization.
- Direct Search Methods of Optimization.
- Basics of Riemannian Differential Geometry.
- Geometric Methods of Optimization using Riemannian Langevin Dynamics.
- Stochastic Analysis on Manifolds and Geometric Optimization Methods.

This textbook comprehensively treats both classical and geometric optimization methods, including deterministic and stochastic (Monte Carlo) schemes. It provides extensive coverage of important topics including derivative-based methods, penalty function methods, method of gradient projection, evolutionary methods, geometric search using Riemannian Langevin dynamics, and stochastic dynamics on manifolds. The textbook is accompanied by online resources including MATLAB codes which are uploaded on our website. The textbook is primarily written for senior undergraduate and graduate students in all applied science and engineering disciplines and can be used as a main or supplementary text for courses on classical and geometric optimization.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Elements of Classical and Geometric Optimization

Debasish Roy and G. Visweswara Rao

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

Designed cover image: Shutterstock

First edition published 2024
by CRC Press
6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742

and by CRC Press
4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

© 2024 Debasish Roy and G. Visweswara Rao

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, access www.copyright.com or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. For works that are not available on CCC please contact mpkbookspermissions@tandf.co.uk

Trademark notice: Product or corporate names may be trademarks or registered trademarks and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data
Names: Roy, Debasish Kumar, 1946– author. | G., Visweswara Rao (Gorti), author.
Title: Elements of classical and geometric optimization / Debasish Roy and
G. Visweswara Rao.
Description: First edition. | Boca Raton : CRC Press, [2023] |
Includes bibliographical references and index.
Identifiers: LCCN 2023003311 (print) | LCCN 2023003312 (ebook) |
ISBN 9780367560164 (hbk) | ISBN 9781032538822 (pbk) | ISBN 9781003414063 (ebk)
Subjects: LCSH: Mathematical optimization. | Manifolds (Mathematics) |
Geometry, Differential. | Engineering mathematics.
Classification: LCC QA402.5 .R69 2023 (print) | LCC QA402.5 (ebook) |
DDC 519.6–dc23/eng20230712
LC record available at https://lccn.loc.gov/2023003311
LC ebook record available at https://lccn.loc.gov/2023003312

ISBN: 978-0-367-56016-4 (hbk)
ISBN: 978-1-032-53882-2 (pbk)
ISBN: 978-1-003-41406-3 (ebk)

DOI: 10.1201/9781003414063

Typeset in Times
by Newgen Publishing UK

Contents

Figures

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

APPENDIX 1

APPENDIX 2

APPENDIX 3

APPENDIX 4

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Tables

Acronyms

ALM	Augmented Lagrangian method
BC	Boundary condition
BFGS	Broyden-Fletcher-Goldfarb-Shanno
BVP	Boundary value problem
CDF	Cumulative distribution function
CG	Conjugate gradient
CGM	Conjugate gradient method
CLT	Central limit theorem
CMA	Covariance matrix adaptation
DE	Differential equation
DFP	Davidon-Fletcher-Powell
DiEv	Differential evolution
div	divergence operator
dof	Degree-of-freedom
EL	Euler Lagrangian
EM	Euler-Maruyama
EQ	Equality
FE	Finite element
FEM	Finite element method
FM	Frame bundle
FIM	Fisher information matrix
GA	Genetic algorithm
GALA	Geometrically Adapted Langevin Algorithm
GE	Greater than or equal to
GL	General linear group
grad	Riemannian gradient
GRG	Generalized reduced gradients
HJ	Hooke and Jeeves
IC	Initial condition
ICh	Incomplete Cholesky
i.i.d	Independent and identically distributed
KKT	Karush-Kuhn-Tucker
KL	Kullback–Leibler
LB	Laplace-Beltrami
LE	Less than or equal to
LHS	Left hand side
LLN	Law of large numbers
LP	Linear programming
LQR	Linear quadratic regulator
MALA	Metropolis Adjusted Langevin algorithm
MC	Monte Carlo
MCMC	Markov chain Monte Carlo

MGF	Moment generating function
MH	Metropolis-Hastings
MLE	Maximum likelihood estimation
NM	Nelder and Mead
NP	Non-deterministic polynomial
ODE	Ordinary differential equation
PDE	Partial differential equation
PSO	Particle swarm optimization
pdf	Probability density function
RBF	Radial basis function
RHS	Right hand side
RMALA	Riemannian version of MALA
RV	Random variable
SA	Simulated annealing
SDE	Stochastic differential equation
SDM	Steepest descent method
SQP	Sequential quadratic programming
TRM	Trust region method
TSP	Travelling salesman problem
ULA	Unadjusted Langevin algorithm

General Notations

$B_t(\omega)$	*Wiener process (Brownian motion) – often denoted in short by* B_t
CDF	*cumulative (probability) distribution function*
C^1, C^2	*continuously differentiable, twice continuously differentiable*
C_0^∞	*space of compactly supported smooth functions that are infinitely differentiable*
E [.]	*expectation operator*
$f(\boldsymbol{x})$	*objective function in the design variable vector,* $\boldsymbol{x}$
$\mathbb{f}_X(x)$	*probability density function, pdf*
$F_X(x)$	*probability (cumulative) distribution function, CDF*
FM_x	*fibre at x on the frame bundle FM*
g	*Riemannian metric*
$\mathfrak{g}$	*matrix associated with the Riemannian metric*
$\boldsymbol{I}$	*identity matrix*
LHS	*left hand side*
$\mathcal{N}(m, \sigma)$	*normal (Gaussian) random variable parametered by* m *and* σ
$\mathbb{N}$	*set of natural numbers*
ODE	*ordinary differential equation pdf probability density function*
$\mathbb{R}$	*set of real numbers (real line)*
$\mathbb{R}^+$	*non-negative real numbers*
$\mathbb{R}^n$	*n-dimensional Euclidean space*
$\mathbb{R}^{n\times m}$	$n \times m$ *matrices with real elements*
RHS	*right hand side*
SDE	*stochastic differential equation*
t	*time variable*
$\mathbb{U}(.)$	*unit step function (Heavyside step function)*
$U(a, b)$	*uniform distribution in the interval* $[a, b]$
$\boldsymbol{x}$	*design variable vector*
$\boldsymbol{x}^*$	*optimum point*
$\mathbb{Z}$	*set of integers*
$\delta(.)$	*Dirac delta function*
δ_i^j, δ_{ij}	*Kronecker delta*

Γ_{ij}^{k}	*Christoffel symbols*
$\|.\|_p$ L_p	*norm*
$\cup$	*union operator*
$\cap$	*intersection operator*

Preface

The broad subject of optimization, an area of widespread usefulness as a modelling tool in science and engineering, can justifiably lay its claim on a vast swathe of literature that prominently includes many books, monographs and articles. In view of this, the natural question to ask is what this book has to offer by way of a novelty or value addition to the potential readership. What kind of preparation should the reader have so they can make the most of this book? We have tried to cover a lot of ground in this book, though the contents are written mostly in a language accessible to an advanced undergraduate student in science and engineering. For instance, this is perhaps the first attempt at a textbook that addresses both classical and geometric aspects of optimization using methods, deterministic and stochastic. However, beyond an elementary course in linear algebra, typically covered during the first or second year of the undergraduate programme and a general flair to learn, we the authors demand little else from a young reader. We claim this despite the rather involved nature of topics covered in Chapters 4 and 5, e.g. elements of the Riemannian metric, connection, exponential and log maps, stochastic calculus and stochastic development. Much of the material covered from Chapter 4 onwards should also be useful for research scholars and scientists desirous of getting a foothold in non-classical optimization techniques grounded in Riemannian geometry and applying them to problems of relevance to their research. Departing from the mathematically rigorous and mostly abstruse pedagogy that mainly dominate the current literature on Riemannian geometric methods, including those dealing with optimization, this book not only strives to present the ideas in a language bearable for scientists and engineers, but takes the initiative a step further by providing algorithms backed up by computer programs (available in the companion website). This is precisely where this book tries to take a positive step at filling in a rock-cleft that has largely been left unattended –dissemination of the powerful concepts in geometric optimization to non-mathematicians.

For details, the precise flow of contents in the book goes as follows. While Chapters 1–3 introduce the problems of optimization and the methods of solution in the classical (i.e. Euclidean) setting, the rest of the book (Chapters 4–5) deals with the elements of geometric optimization with an emphasis on methods based on stochastic search. Chapter 1 is introductory and highlights the ubiquitous role of optimization, i.e., to search for an optimum – maximum or minimum. Starting with a mathematical posing of unconstrained/constrained optimization problems, we attempt in the chapter to introduce the necessary and sufficient conditions underlying an optimal solution – in a language simple enough even for the beginner. Amidst the varied and centuries-old growth in the solution methods of optimization, this chapter tracks the emergence of variational calculus which is foundational to much of science and engineering. We also try to highlight the vital link between probability theory and the meta-heuristic/evolutionary methods of optimization that form the central theme of a later chapter. We have included illustrations via graphical presentation wherever applicable, to enable the reader to develop a clearer perspective on the concepts presented.

In Chapter 2, we carry forward these basic notions underlying any typical optimization problem and initiate a graded presentation of the classical methods of solution, especially the derivative-based ones. Keeping in view the significant role of unconstrained optimization methods in solving problems even with constraints, we stress the innovative concepts underlying each of these techniques. The presentation is followed by a mix of examples and we illustrate in the sequel the applicability of these methods to different scenarios where the objective function may be explicit or otherwise.

An obvious difficulty with derivative-based methods is the evaluation of derivatives, which has been inspiration enough for the emergence of many derivative-free schemes. Chapter 3 engages with these methods, also called direct-search methods. While we narrate the pattern search and trust region strategies as some of the popular and robust techniques belonging to this category, we also highlight the significant place occupied by the evolutionary methods. These are special techniques known as stochastic search methods and characterized by an underlying probability model and random sampling that enables efficient exploration of the search space for locating optimal solutions at each iteration. By this stage, we expect our reader to have come to terms with the notion that (i) all optimization solution methods are iterative and more so for nonlinear constrained problems and that (ii) unconstrained optimization methods may be hybridized with Lagrangian multipliers or penalty functions (described in Chapters 1 and 2) to treat problems with constraints.

Chapter 4 concerns the geometric methods of optimization. These are based on Riemannian differential geometry. In the backdrop of a vast literature and given the umbilical cords geometric methods have with myriad fields, what we have put together is no more than an apology for the most essential aspects. A curved hypersurface or a manifold is the fundamental object of differential geometry. Our interest is to search for the optimum on a manifold, which may appear implicitly through the objective function or through the prescribed constraints. First, we have tried to familiarize the reader with elements of Riemannian geometry, i.e. with the locally Euclidean property, differentiability, coordinate charts, the metric, the affine connection, the geodesic and the exponential map. This should have enabled the reader to understand the steps in a makeover of the classical methods of optimization to their manifold versions. Such algorithms are provided to showcase the improved performance of the manifold versions of some of the descent methods. Similar is the case with the statistical estimation problem solved in Chapter 3 by classical methods. We also discuss stochastic search methods described in Chapter 3 and suited for non-convex optimization. The presentation is influenced by the analogy of optimization methods and statistical sampling by Markov chain Monte Carlo (MCMC). This analogy leads to optimization algorithms that are intrinsically stochastic; here the design variables evolve through a stochastic differential equation (SDE) known as the Langevin SDE and the solution is brought back to the manifold through exponential mapping. This also brings up the need to acquaint the reader with a fair understanding of stochastic processes. Having done this, we discuss the stochastic optimization method based on Langevin dynamics in the last section of this chapter. Of course, the technique has

the limitation that the Brownian motion in the Langevin SDE is not designed to live on the manifold.

This brings us to Chapter 5, where we discuss stochastic development that enables us to intrinsically transfer solutions of SDEs posed in the Euclidean setting to a Riemannian manifold of the same dimension. The transfer is intrinsic in that the Riemannian manifold need not be embedded within a higher dimensional Euclidean space. A special case of this stochastic development is a Brownian motion restricted to evolve on a manifold. To exploit this concept for non-convex optimization, we again make use of the Langevin dynamics whose solution is stochastically developed on a Riemannian manifold defined by the objective function. The results by this geometric scheme for a few selected test functions indicate faster convergence and higher accuracy.

As previously noted, the work undertaken in this book is somewhat ambitious as we try to conjure up the essence of a few advanced ideas, all implemented in the context of optimization, with a language simple enough to be grasped by an advanced undergraduate student. Equally challenging has been the task of motivating the reader to this onerous task, starting with the very elements of classical optimization methods. These lofty goals notwithstanding, we may not have quite succeeded in meeting them. It is here that suggestions from the reader on a future improvement of the presentation and contents are most important (just send an email to the first author at royd@iisc.ac.in or royd.civil@gmail.com). In this context, we are grateful to Dr. Mariya Mamajiwala, a former student of the first author, who had painstakingly gone through an earlier version of the manuscript and suggested several useful modifications. The first author also wishes to thank his student, Mr. Ankit Tyagi, for help with formatting the manuscript files as required by the publisher. Finally, we would feel happy if, despite all its shortcomings, our readers and reviewers feel that the book has taken a small step in realizing the stated aims.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Author Biographies

Debasish Roy, Professor, Department of Civil Engineering, Indian Institute of Science, Bangalore obtained his Ph.D. from the Indian Institute of Science, followed by post-doctoral research at the University of Innsbruck, Austria. He has published over 140 research papers in journals of national and international repute. His current research areas include geometrically inspired and gauge theories for continuum mechanics of solids, non-equilibrium thermodynamics of solids and fluctuation relations valid far from equilibrium, defect engineering and metamaterials with acoustic band gaps, and optimization based on stochastic search on Riemannian manifolds.

G. Visweswara Rao is currently working as an engineering consultant in Bangalore, India. He received his Ph.D. from the Indian Institute of Science, Bangalore, in 1989. He has published several research papers in the areas of structural dynamics specific to earthquake engineering, nonlinear and random vibration, and structural control. His areas of research include non-linear and stochastic structural dynamics.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

1 Optimization Methods
A Preview

1.1 INTRODUCTION

Optimization problems appear routinely, rather than as exceptions, in all fields of science, engineering, finance, management and industry (Nicholson 1971, Papadimitriou and Steiglitz 1982, Goldberg 1989, Cornuejols and Tütüncü 2007). Optimizing some measure of cost or effort or time expended in realizing an objective is perhaps a principle not only guiding the evolution of nature's species, but also motivating the short-term response of an individual or a group. It could be posed in a continuous or discrete setting, even though this book deals mainly with the former. A familiar example of continuous optimization is the curve-fitting problem – i.e. fitting data, either observed or experimentally extracted, to a continuous function. This could be done, among other means, by least-squares error minimization. As another example, investors in financial circuits always look to maximize their profit with minimum risk. At the microeconomics level, a consumer's interest is in maximizing, say, the utility function $\mathcal{U}(\boldsymbol{x})$, which is continuous, and quantifies the utility of the goods purchased. Let us stick with this example a little longer to get a feel for how to pose an optimization problem and how a solution may be graphically obtained for some simple cases. Here $\boldsymbol{x}$ is the vector of control or design variables. It is an n-dimensional vector, whose elements consist of only positive numbers representing the quantity of goods that one would like to purchase. The obvious constraint in the problem is the amount $\mathcal{A}$ available with the buyer for the expenditure. It is a constrained optimization problem which can be stated as:

$$\text{maximize } \mathcal{U}(\boldsymbol{x}) \tag{1.1a}$$

$$\text{s.t. } g(\boldsymbol{x}) = \boldsymbol{c}^T \boldsymbol{x} \leq \mathcal{A} \tag{1.1b}$$

's.t.' is an abbreviation for 'subjected to' or 'such that'. $g(\boldsymbol{x})$ denotes the constraint in the form of maximum permissible expenditure. $\boldsymbol{c} \in \mathbb{R}_+^{\boldsymbol{n}}$ is the vector of prices corresponding to $\boldsymbol{x}$. Notation-wise, $\mathbb{R}^{\boldsymbol{n}}$ represents an n-dimensional Euclidean space and $\mathbb{R}_+^{\boldsymbol{n}}$ denotes the first quadrant in the n-dimensional space. Thus $\boldsymbol{c} \in \mathbb{R}_+^{\boldsymbol{n}}$ implies that all its elements are positive real numbers. The design space Ξ is the set in which the design variable $\boldsymbol{x}$ take values and, in this example, $\Xi = \mathbb{R}_+^{\boldsymbol{n}}$. As an illustration, let $n = 2$ with $\boldsymbol{x} = (x_1, x_2)^T$, $\boldsymbol{c} = (c_1, c_2)^T$, $\mathcal{U}(\boldsymbol{x}) = \sqrt{x_1 x_2}$ and $\Xi = \mathbb{R}_+^2$ (represented by the $x_1 - x_2$ plane). Here the superscript 'T' denotes transposition. From practical considerations,

DOI: 10.1201/9781003414063-1

most optimization problems are in general constrained like the present example. While various solution methods exist to handle optimization problems – to be progressively discussed in this chapter and the ones that follow in this book – we particularly utilize this example to demonstrate that one may arrive at the optimum $\boldsymbol{x}^*$ by means of a simple geometrical construction.

Let the parameters be given by $c_1 = 3, c_2 = 1$ and $\mathcal{A} = 400$. In Figure 1.1, different contours of $\mathcal{U}(\boldsymbol{x}) = \alpha$ are drawn with α^2 varying in the interval [5000, 20000]. Restricted to graphs in the first quadrant of the $x_1 - x_2$ plane, these contours correspond to $x_1 x_2 = \alpha^2$ and are rectangular hyperbolas and represent the equi-potential (utility) curves. The constraint Equation (1.1b) appears as a straight line AB in the plane. The set of all possible points in the design space Ξ that satisfy the specified constraints is known as the feasible space $\mathcal{D} \subset \Xi$ and is given by the region below the line AB in the first quadrant. The optimum is traced to the point $\boldsymbol{x}^*$ on a contour $\mathcal{U}(\boldsymbol{x}) = \alpha^*$ closest to the line AB. The optimum point – where AB is a tangent to one of the hyperbolas – is approximately (66.5, 200). At this point, the utility function attains its maximum value given by $\alpha^* \cong 115.3$ with $\alpha^{*2} \cong 13300$.

We may introduce a variation in the optimization problem by modifying the statement in Equation (1.1) to formulate a minimization problem. Consider the

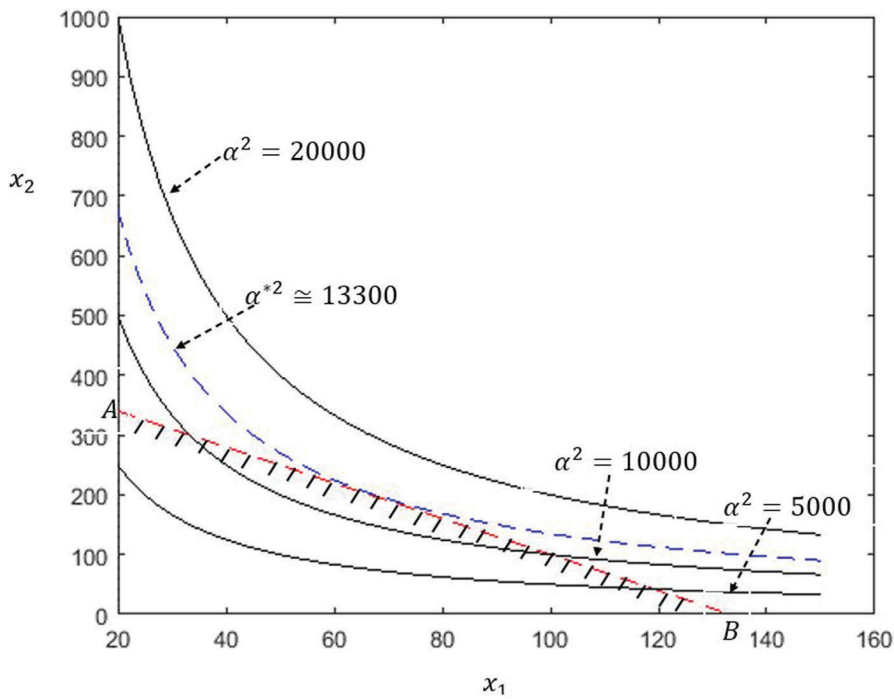


FIGURE 1.1 Maximization of the utility function $\mathcal{U}(x) = \sqrt{x_1 x_2} = \alpha$; graphical solution, straight line AB represents the limiting constraint $\boldsymbol{c}^T \boldsymbol{x} = \mathcal{A} = 400$, feasible region – region in the first quadrant below AB (hatched in the figure)

cost of production in a factory where $\boldsymbol{x}$ represents a vector of quantities of different input materials and $\boldsymbol{c}$, the corresponding vector of investment costs. In this case, $g(\boldsymbol{x}) = \sqrt{x_1 x_2}$ may act as a limiting constraint in the form of the minimum required produce (output) from the factory. We may state the optimization problem as:

$$\text{minimize } f(\boldsymbol{x}) = \boldsymbol{c}^T \boldsymbol{x} \tag{1.2a}$$

$$\text{s.t. } g(\boldsymbol{x}) = \sqrt{x_1 x_2} \geq \mathcal{A} \tag{1.2b}$$

For illustration, the investment costs c_1 and c_2 are assumed to be 4 and 1, respectively. Let the firm be expected to produce at least 100 units, i.e. $\mathcal{A} = 100$. Figure 1.2 shows the graphical solution with $\boldsymbol{x}^* = (50, 200)$ and $f(\boldsymbol{x}^*) = \boldsymbol{c}^T \boldsymbol{x}^* = \alpha = 400$. In this case, the feasible region $\mathcal{D}$ satisfying $g(\boldsymbol{x}) = \sqrt{x_1 x_2} \geq \mathcal{A}$ lies above the rectangular hyperbola in the first quadrant of the $x_1 - x_2$ plane.

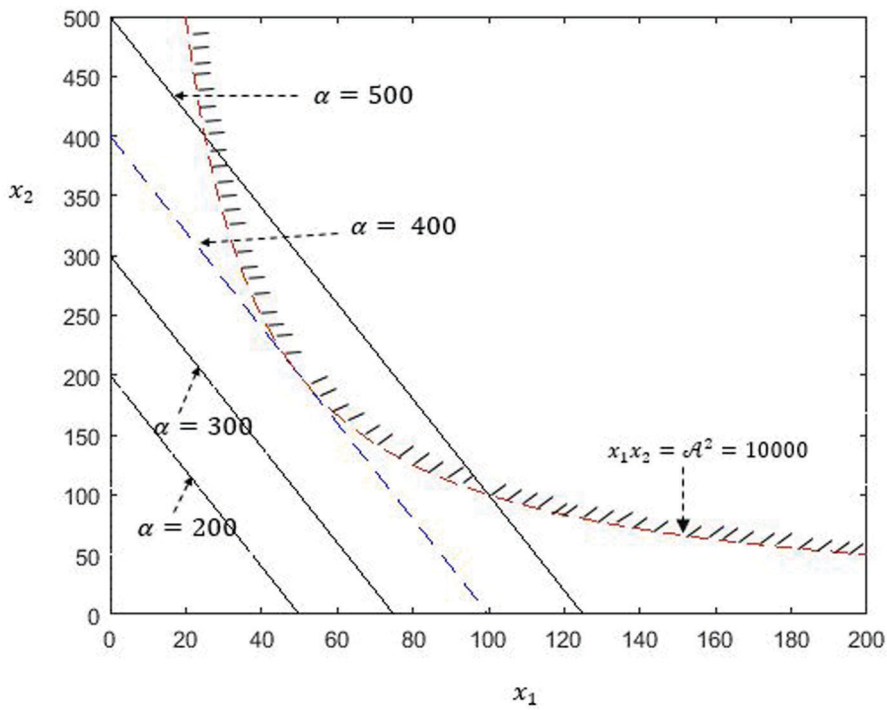


FIGURE 1.2 Minimization of the capital cost $f(x) = \boldsymbol{c}^T x$ where $\boldsymbol{c} = (4,1)^T$; graphical solution, straight lines correspond to the equi-cost curves of $\boldsymbol{c}^T x = \alpha$, hyperbola represents the limiting constraint function $x_1 x_2 = \mathcal{A}^2 = 10000$, feasible region – region in the first quadrant above the hyperbola (hatched in the figure)

A similar way of posing an optimization problem may be found across myriad scientific and vocational streams – be it resource management, process planning or system design. For instance, resource management can be thought of as a task aimed at an efficient utilization of the available resources – from the point of investment and its utilization. A familiar problem is that of vehicle routing (Dantzig and Ramser 1959, Bodin *et al.* 1983, Laporte 1992a, Cordeau *et al*. 2002, Govindan *et al*. 2018, Vahdani and Shahramfard 2019) and freight scheduling (Dulebenets 2018, 2019), which are related to service delivery within supply chains. Specifically, it is a case of minimizing the mileage of vehicles allotted to deliver goods from depots to different stations at specified locations.

Discrete optimization problems are also posed in a similar manner. A few examples include computer wiring (Lenstra and RinnooyKan 1975), job sequencing (Garfinkel 1985) and circuit board drilling (Reinelt 1994). Printed circuit board (PCB) drilling is an important exercise that automates the process of drilling holes on a circuit board in minimum time. Since the holes may be of different diameters, it makes sense to so order the drilling sequence that holes of the same diameter are finished before the machine is programmed to go on to the next set of holes of a different size. Paradigmatic of all such discrete or combinatorial optimization problems is the travelling salesman problem (TSP) (Lawler *et al.* 1985, Laporte 1992b). TSP is the problem of minimizing the distance travelled by a salesperson visiting a given set of cities with each city visited only once, before returning to the starting one. TSP is generally formulated by a graph theoretic approach using a complete graph $G = (\boldsymbol{V}, \boldsymbol{E})$ (Figure 1.3) with the nodes $\boldsymbol{V}$ representing cities and $\boldsymbol{E} := \{E_{ij}, i = 1,2,\ldots,N, j = 1,2,\ldots,N\}$ the edges in the graph. A complete graph is one in which each pair of vertices is joined by a unique edge. In graph theoretic parlance, one attempts at finding the shortest Hamiltonian cycle in TSP. A Hamiltonian cycle passes through each node exactly once and ends at the starting node. Figure 1.4 shows some possible Hamiltonian cycles in a five-noded graph.

TSP occupies a unique place among optimization problems in that it has many applications in operations research and game theory (von Neumann 1928, von

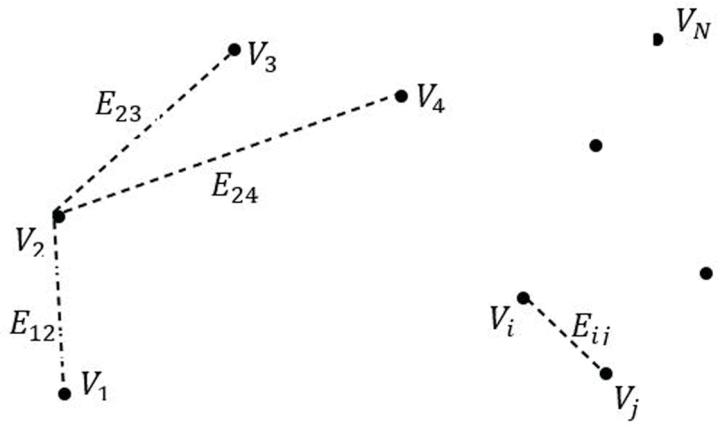


FIGURE 1.3 TSP; $V_i, i = 1, 2, \ldots, N$ represent cities and $E_{ij}, i = 1, 2, \ldots, N,\ j = 1, 2, \ldots, N$ represent edges between V_i and V_j.

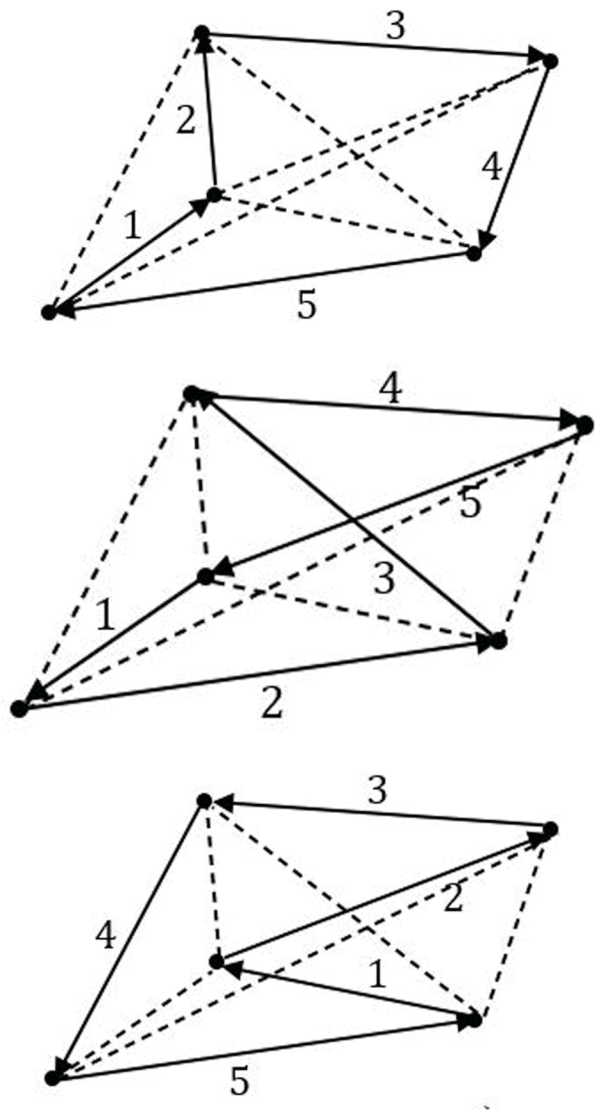


FIGURE 1.4 a–c Some Hamiltonian cycles 1–2–3–4–5 (in dark line with arrows) in a five-noded complete graph.

Neumann and Morgenstern 1953, Rardin 1997, Curiel 1997, Gutin and Punnen 2002, Derigs 2009). In fact, the circuit board drilling problem may be considered as an application of a sequence of TSPs, with each one corresponding to holes of the same diameter.

As already stated, this book is aimed primarily at methods of continuous optimization. Just as TSP is paradigmatic of discrete optimization problems, the brachistochrone problem in mechanics plays a similar role in continuous optimization

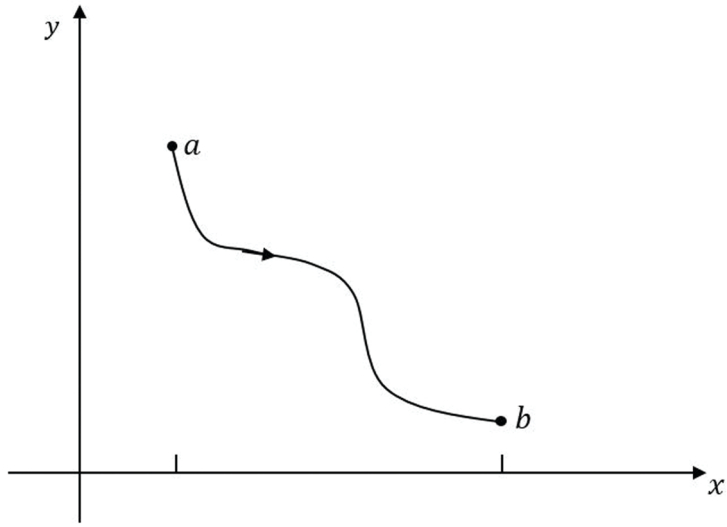


FIGURE 1.5 Brachistochrone problem; a typical path $y(x)$ between the points a and b.

(Bernoulli 1697, Courant 1943, TerHaar 1971). The brachistochrone problem is used to determine the shape of the curve for which the time of descent of a bead under gravity from point a to b (see Figure 1.5) is a minimum. Bernoulli, who originally posed the problem to the world in 1696, himself offered (1697) an ingenious solution based on the optical analogy of Fermat's principle of least time (Erlichson 1999). The problem was solved independently by Leibniz, Newton and L'Hospital using the 'variational principle' or the principle of least action involving minimization of a 'functional'. This, in fact, led to the emergence of variational calculus as a powerful tool for deriving the conservation laws in various fields of science and engineering.

The brachistochrone problem is thus the main precursor to later developments in variational calculus and the ensuing growth in all branches of mechanics. In fact, the variational principles of mechanics provided the impetus for developing the finite element method (FEM) (Hrenniff 1941, Courant 1943, Strang and Fix 1973, Zienkiewicz 1977, Bathe 1996) as a fundamental and indispensable numerical tool in engineering analysis and design. Application areas of the FEM include solid and structural mechanics (Reddy 2002, Cassel 2013, Dym and Shames 2013), fluid dynamics (Chorin and Marsden 1993, dell'Isola and Gavrilyuk, 2011), electromagnetics (Jackson 1999, Garg 2008), thermodynamics (O'Connell and Haile 2005, Basdevant 2007) and control engineering (Troutman 1996, Liberzon 2012). Section 1.6 briefly highlights the umbilical connection that the origin of the FEM has with variational calculus.

The two archetypal problems – TSP and brachistochrone – and the early attempts to find their solutions will be described shortly. With the current emphasis mostly on continuous optimization problems, we now give a more formal shape to the definitions in Equations (1.1) and (1.2) before briefly coming back to the TSP in Section 1.3.

1.2 THE CONTINUOUS CASE – MATHEMATICAL FORMULATION

Any optimization problem in a general setting reads as follows.

Given an n-dimensional design space Ξ, minimize (or maximize) the objective/cost function:

$$f(\boldsymbol{x}): \mathbb{R}^n \to \mathbb{R} \tag{1.3a}$$

$$\text{s.t.}\ \boldsymbol{h}(\boldsymbol{x}) = 0,\ \boldsymbol{g}(\boldsymbol{x}) \le 0, \text{and}\ \boldsymbol{x}_L \le \boldsymbol{x} \le \boldsymbol{x}_u \tag{1.3b}$$

where $\boldsymbol{x} \in \Xi$ are the design variables. For the present, assuming that f is a single objective function, we say that f maps $\mathbb{R}^n$ to $\mathbb{R}$. Thus the domain of f (which is same as the range of $\boldsymbol{x}$) is $\mathbb{R}^n$ and its range is $\mathbb{R}$. $\boldsymbol{h}: \mathbb{R}^n \to \mathbb{R}^l$ and $\boldsymbol{g}: \mathbb{R}^n \to \mathbb{R}^m$ are commonly referred to as equality and inequality constraint vectors of dimension l and m, respectively. The vectors $\boldsymbol{x}_L$ and $\boldsymbol{x}_u$ are, respectively, the lower and upper bounds for the design variables $\boldsymbol{x}$. These constraints fix the feasible region $\mathcal{D}$ in the design space (see Figures 1.1 and 1.2 for an illustration of feasible regions).

Weierstrass theorem for the optimum of a function
If $f(\boldsymbol{x})$ is real-valued continuous on a non-empty compact domain $\mathcal{D}$, then there exists an $\boldsymbol{x}^* \in \mathcal{D}$ that minimizes (or maximizes) $f(\boldsymbol{x})$ on the set $\mathcal{D}$.

For an unconstrained optimization problem, the feasible space $\mathcal{D}$ is the same as the design space Ξ. Clearly, an optimal solution is possible only if $\mathcal{D}$ is non-empty. A domain is compact if it consists of compact sets. A set is compact if it is both closed and bounded. The notions of closed and open sets complement each other, i.e. a closed set is one which is not open. Suppose that $\mathcal{D} = \mathbb{R}$. In mathematical terms, a subset $A \subseteq \mathbb{R}$ is open if $\exists \varepsilon > 0$ such that $(x - \varepsilon, x + \varepsilon) \subseteq A$, $\forall x \in A$. Essentially open intervals on the real line are open sets. In higher dimensions, we may consider an open ball in $\mathbb{R}^n$, $\mathcal{B}_r(\boldsymbol{x}_0) = \{\boldsymbol{x} : |\boldsymbol{x} - \boldsymbol{x}_0| < r\}$ with centre $\boldsymbol{x}_0$ and radius $r > 0$. For example, the set of real numbers $(1 < x < 3) \in \mathbb{R}$ is an open set and thus is not closed. So if $\mathcal{D} = (1 < x < 3)$, there need not be an optimal solution since $\mathcal{D}$ is not closed. Similarly, according to the Weierstrass theorem, no optimal solution is possible when $\mathcal{D} = (x \ge 0)$ since $\mathcal{D}$ is not bounded.

1.2.1 Unconstrained Optimization and Optimality Conditions

With $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$, consider the optimization of $f(\boldsymbol{x})$ with no constraints. In this case, a familiar criterion for an optimum (minimum or maximum) is the requirement of stationarity. Stationary points of a differentiable function $f(\boldsymbol{x})$ are the points where its (the function's) first-order partial derivatives are zero or the 'gradient' $\nabla f(\boldsymbol{x}) := \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right)^T \in \mathbb{R}^n$ is zero, i.e. the zero vector in $\mathbb{R}^n$. This is a necessary condition for $\boldsymbol{x}^*$ to be an optimum point. If one considers the matrix $\boldsymbol{H}(\boldsymbol{x})$(denoted by $\boldsymbol{H}$ for brevity) of second-order derivatives (known as the Hessian matrix):

$$\boldsymbol{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ & \vdots & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \tag{1.4}$$

then, the sufficiency condition for $\boldsymbol{x}^*$ to be a minimum is that $\boldsymbol{H} \in \mathbb{R}^{n \times n}$ be positive definite. On the other hand, the condition for $\boldsymbol{x}^*$ to be a maximum is that $\boldsymbol{H}$ be negative definite. Note that $\boldsymbol{H}$ is symmetric. Positive definiteness (or negative definiteness) of $\boldsymbol{H}$ is associated with the nature of the quadratic function $\boldsymbol{x}^T \boldsymbol{H}\boldsymbol{x}$. Indeed, for any n, $\boldsymbol{x}^T \boldsymbol{H}\boldsymbol{x} \in \mathbb{R}$, i.e. it assumes a scalar value for each $\boldsymbol{x}$. If $\boldsymbol{x}^T \boldsymbol{H}\boldsymbol{x} > 0$ for all $\boldsymbol{x}$, $\boldsymbol{H}$ is said to be positive definite and if $\boldsymbol{x}^T \boldsymbol{H}\boldsymbol{x} < 0$, $\boldsymbol{H}$ is negative definite. The condition $\boldsymbol{x}^T \boldsymbol{H}\boldsymbol{x} > 0$ for a minimum is equivalent to $f(\boldsymbol{x})$ being locally quadratic around and strictly convex at $\boldsymbol{x}^*$ (see Figure 1.6 for the definition of a convex function).

On the other hand, $f(\boldsymbol{x})$ being locally quadratic and strictly concave (with an opposite meaning to convexity) at maximum $\boldsymbol{x}^*$ is ensured by the sufficiency condition $\boldsymbol{x}^T \boldsymbol{H}\boldsymbol{x} < 0$.

In the one-dimensional case ($n = 1$), $x^T Hx = x^2 \dfrac{d^2 f}{dx^2}$ and the above two optimality (necessary and sufficient) criteria are respectively (i) $\left.\dfrac{df}{dx}\right|_{x=x^*} = f'(x^*) = 0$ and

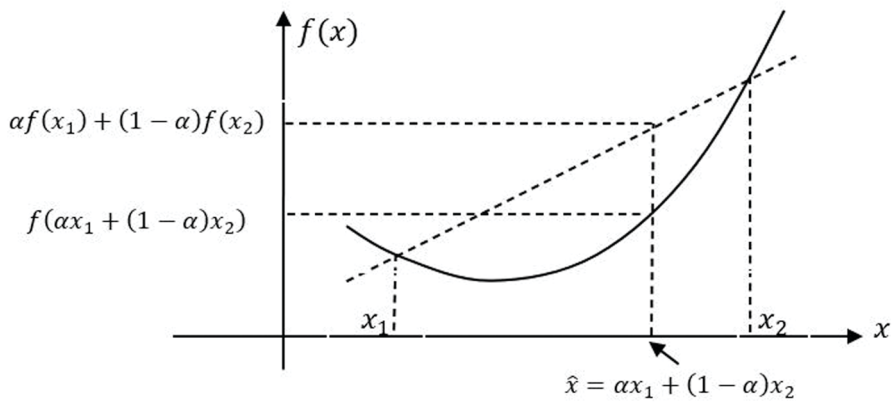


FIGURE 1.6 A function $f(x)$ is convex if $f(\hat{x}) \le \alpha f(x_1) + (1-\alpha) f(x_2)$ for any $\alpha \in [0,1]$; strictly convex if the inequality sign always holds.

(ii) $\left.\frac{d^2 f}{dx^2}\right|_{x=x^*} > 0$ for a minimum and $\left.\frac{d^2 f}{dx^2}\right|_{x=x^*} < 0$ for a maximum. In this case, it is easy to see that the sufficient condition $\left.\frac{d^2 f}{dx^2}\right|_{x=x^*} > 0$ for a minimum relates to change of slope $\frac{df}{dx}$, of the objective function from –ve to +ve as it crosses x^* – from left to right, a characteristic feature of a convex function. For a maximum, it is just the opposite with $\left.\frac{d^2 f}{dx^2}\right|_{x=x^*} < 0$.

In fact, any continuous function $f(\boldsymbol{x})$ whose first two derivatives exist can be locally approximated by a quadratic function $\hat{f}(\boldsymbol{x})$ around any $\hat{\boldsymbol{x}}$ using a truncated Taylor series expansion as:

$$f(\boldsymbol{x}) \cong \hat{f}(\boldsymbol{x}) = f(\hat{\boldsymbol{x}}) + \nabla f(\hat{\boldsymbol{x}})^T (\boldsymbol{x} - \hat{\boldsymbol{x}}) + \frac{1}{2}(\boldsymbol{x} - \hat{\boldsymbol{x}})^T \boldsymbol{H}(\hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}}) \tag{1.5}$$

One of the oft-quoted functions in the optimization literature, the Rosenbrock function $f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2$ is shown in Figure 1.7a. It is easy to verify that $x_1 = x_2 = 1$ is the optimum point $\boldsymbol{x}^*$ with the function value $f(\boldsymbol{x}^*) = 0$. It is also verifiable that the necessary condition is satisfied at $\boldsymbol{x} = \boldsymbol{x}^*$:

$$\nabla f\big|_{\boldsymbol{x}=\boldsymbol{x}^*} = \left.\begin{pmatrix} 400(x_1^2 - x_2)x_1 + 2(x_1 - 1) \\ -200(x_1^2 - x_2) \end{pmatrix}\right|_{\boldsymbol{x}=\boldsymbol{x}^*} = \{\boldsymbol{0}\} \tag{1.6}$$

The Hessian matrix is given by:

$$\boldsymbol{H} = \begin{bmatrix} 1200x_1^2 - 400x_2 + 2 & -400x_1 \\ -400x_1 & 200 \end{bmatrix} \tag{1.7}$$

In this case, $\boldsymbol{x}^T \boldsymbol{H} \boldsymbol{x}$ is given by $H_{11}x_1^2 + 2H_{12}x_1x_2 + H_{22}x_2^2$. Note that $\boldsymbol{x}^{*T} \boldsymbol{H} \boldsymbol{x}^* = 202 > 0$ showing that $\boldsymbol{x}^* = (1,1)^T$ is a minimum point for the Rosenbrock function. In the vicinity of $\boldsymbol{x}^*$, one can approximate the Rosenbrock function as a quadratic function $\hat{f}(\boldsymbol{x})$ by using Equation (1.5) as $\hat{f}(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^*)^T \boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}^*)$. Figure 1.7b shows $\hat{f}(\boldsymbol{x})$ for $\boldsymbol{x} \in (\boldsymbol{x}^* - \boldsymbol{\varepsilon}, \boldsymbol{x}^* + \boldsymbol{\varepsilon})$ with, say, $\boldsymbol{\varepsilon} = (0.05, 0.05)^T$.

Recall that the optimality conditions above are applicable only to an unconstrained problem; we discuss the relevant conditions for a constrained problem in Section 1.7. We may mention that many of the methods – particularly the derivative-based – that solve a constrained optimization problem transform the problem into an unconstrained one at each iteration step. The brachistochrone problem for which a solution

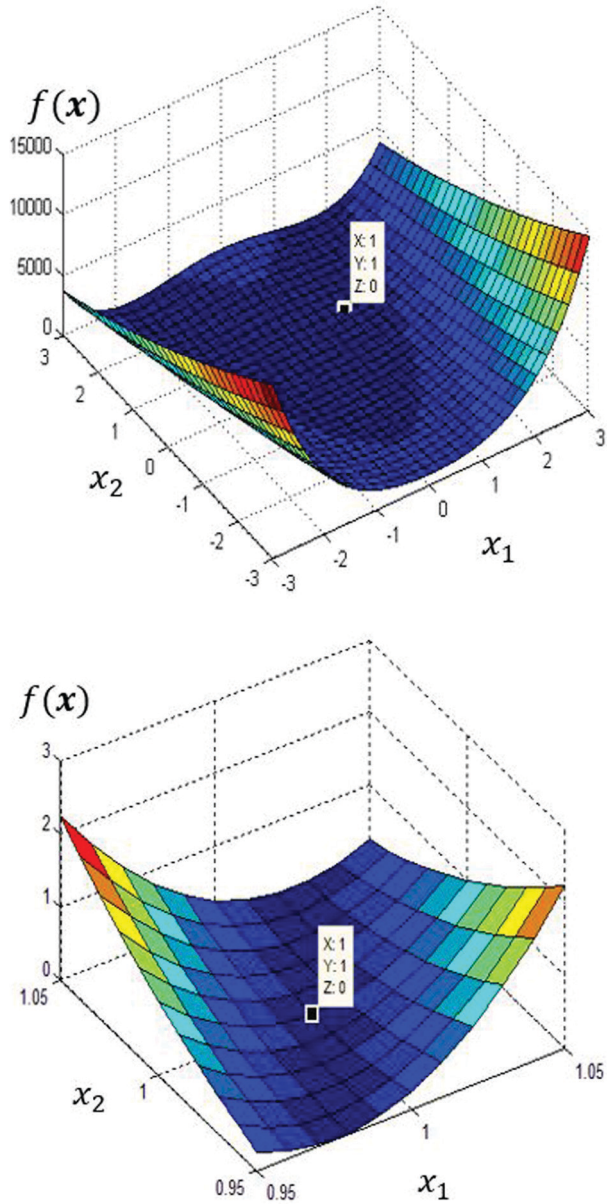


FIGURE 1.7 (a) Rosenbrock function: $f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2$; (b) locally quadratic (convex) function approximated by Equation (1.5) at the minimum point $\boldsymbol{x}^* = (1, 1)^T$ for $\boldsymbol{x}_1 \in (0.95, 1.05)$ and $\boldsymbol{x}_2 \in (0.95, 1.05)$.

(Section 1.5) was offered by Newton and Leibniz using variational calculus in the late 16th century, uses the optimality condition of vanishing 'functional derivative'. This is similar to the necessary condition for optimality of an unconstrained optimization problem.

The travelling salesman problem, the other classic example of optimization is discussed in the next section. The discussion is first on obtaining a solution via a simple brute force technique of direct search. In what follows, we present an alternative and effective strategy founded on probabilistic concepts and what is generally known as Monte Carlo simulation (described in Section 1.4 below). In the sequel, we attempt to draw an analogy to statistical thermodynamics (Binder 1997) and arrive at a true or global optimum to the TSP. A distinction between a 'local' and 'global' extremum is elucidated during the discussion.

1.3 THE DISCRETE CASE – TRAVELLING SALESMAN PROBLEM

Our main interest in discussing the TSP here is to give the reader a preview of how an optimization method can benefit when posed within a probabilistic (or stochastic) setting. The TSP being discrete, we do not presently need a more advanced mathematical language for continuous and diffusive stochastic processes to describe this technique.

In TSP, the objective is to find the shortest tour through a given set of cities using the concept of a Hamiltonian cycle. If each edge E_{ij} (Figure 1.3) is marked with a weight w_{ij} denoting its distance or length in appropriate units, the mathematical formulation of the minimization problem (Dantzig *et al.* 1954) is as follows.

$$\text{Minimize } f = \sum_{i,j}^{N} w_{ij} x_{ij} \text{ over a Hamiltonian cycle} \tag{1.8}$$

where N is the number of cities. The design variables $x_{ij}, i = 1,2,\ldots,N$ and $j = 1,2,\ldots,N$ are binary variables and each is equal to 1 if the edge E_{ij} is included in the tour and zero if it is not. The constraints are:

$$\sum_{j \neq i} x_{ij} = 1, \ \forall i = 1,2,\ldots,N \tag{1.9a}$$

$$\sum_{j \neq i} x_{ij} = 1, \ \forall j = 1,2,\ldots,N \tag{1.9b}$$

Since x_{ij} is a binary variable, constraints in Equation (1.9) are to ensure that each city is touched just once. Also, in the optimization process, care should be taken to avoid sub-tours. Figure 1.8 shows two such sub-tours (not qualifying as a Hamiltonian cycle and thus not a proper tour) in a network of cities with $N = 6$.

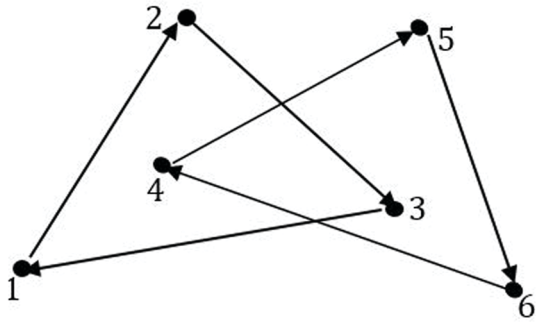


FIGURE 1.8 Sub-tours in a network of six cities ($N = 6$).

TABLE 1.1
Steps in a Brute-force Solution to the TSP

1. Start with an arbitrary node V_i, $i \in [1, N]$ and label it as the current one.
2. Trace out the nearest node $V_j \in [1, N] \setminus i$ connecting the current one which must be an unvisited node.
3. Label V_j as the current node and add w_{ij} to the distance travelled so far.
4. If all the nodes are visited (thus completing a Hamiltonian cycle), store the total distance travelled.
5. Start a new tour with another permutation of the nodes and follow steps 3 and 4.
6. Finish all the permutations and find the tour with the least travel distance.

1.3.1 Brute-force Solution to the TSP

Consider a solution to the TSP by a straightforward direct search. The steps are described in Table 1.1.

The total number of trial tours explored by the brute-force technique is $(N-1)!$ if the starting city is fixed, otherwise it is $N!$. This is the case when the search is unable to distinguish the possible circular permutations. Note that with $N = 12$, maximum number of the trial tours is an incredibly large number: 479001600 and the method quickly becomes computationally prohibitive with increasing N. Now, as an example, consider a TSP with the N cities located on a circular curve, perhaps not sequentially. Note that the optimum tour must be the polyhedron passing through the cities. As $N \to \infty$, the optimum travel distance tends to the circumference of the circle. Figure 1.9 shows one such optimum obtained by the brute-force approach with 12 cities located on a unit circle.

Consider a more realistic example of $N = 12$ cities arbitrarily located in the $X - Y$ plane with the distance matrix given in Table 1.2.

Figure 1.10 shows the shortest Hamiltonian cycle obtained by the brute-force technique resulting in a minimum tour distance of 1778 units.

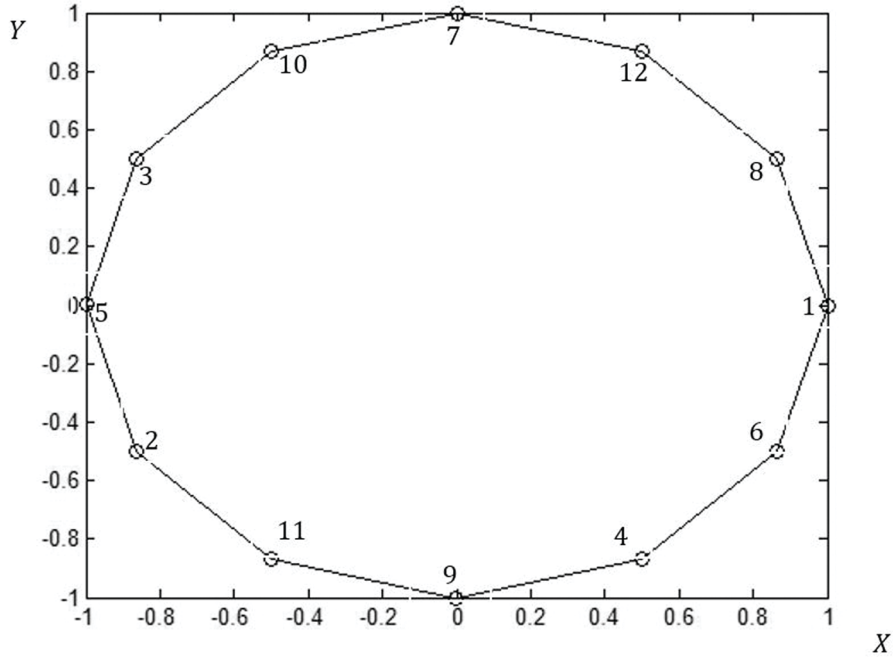


FIGURE 1.9 Brute-force solution to the TSP; $N = 12$ cities (spread not in a sequential order) on a unit circle; optimum distance travelled = 6.21 units (as against the correct value of 6.28).

TABLE 1.2
Distance Matrix of 12 Cities

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	447	380	86	209	403	154	265	213	150	245	295
2	447	0	306	472	242	163	351	416	586	339	277	192
3	380	306	0	344	240	146	400	172	397	231	143	350
4	86	472	344	0	231	397	231	202	135	138	228	346
5	209	242	240	231	0	206	164	235	352	107	107	142
6	403	163	146	397	206	0	364	283	486	261	171	258
7	154	351	400	231	164	364	0	348	364	197	258	168
8	265	416	172	202	235	283	348	0	226	152	144	377
9	213	586	397	135	352	486	364	226	0	248	318	477
10	150	339	231	138	107	261	197	152	248	0	96	244
11	245	277	143	228	107	171	258	144	318	96	0	241
12	295	192	350	346	142	258	168	377	477	244	241	0

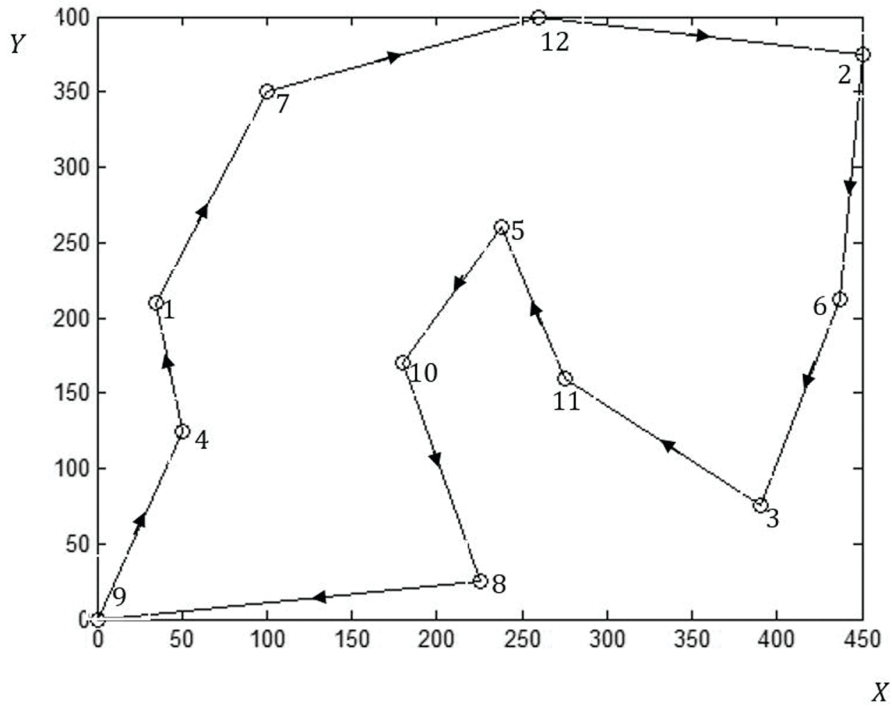


FIGURE 1.10 Brute-force solution to TSP; $N = 12$, computed optimum tour distance = 1778 units, the shortest Hamiltonian cycle is 11 – 5 – 10 – 8 – 9 – 4 – 1 – 7 – 12 – 2 – 6 – 3 – 11.

TABLE 1.3
Solution to TSP Corresponding to N=12 Cities with Distance Matrix in Table 1.2; the Execution Times by Brute-Force Technique (on laptop of version I7 with 8 GB RAM)

N	4	5	6	7	8	9	11	12
Execution time in secs.	0.15	0.17	0.19	0.20	0.25	0.9	93.5	1514

As stated earlier, with increase in size N of the TSP, computational effort increases exponentially and the brute-force technique simply becomes unmanageable (Fogel 1988). Table 1.3 shows the execution time by this technique for a TSP with N varying from 4 to 12. As shown in the table, the computational complexity (Papadimitriou and Yannakakis 1991, Papadimitriou 1994) of solving even a moderate-sized TSP rapidly increases. As regards computational complexity, TSP is an NP-hard optimization problem and if posed as a decision problem it is NP-complete. While an optimization problem is about finding an extremum, a decision problem seeks an answer: 'yes' or 'no'. A TSP becomes a decision problem if we ask the question "Does a Hamiltonian

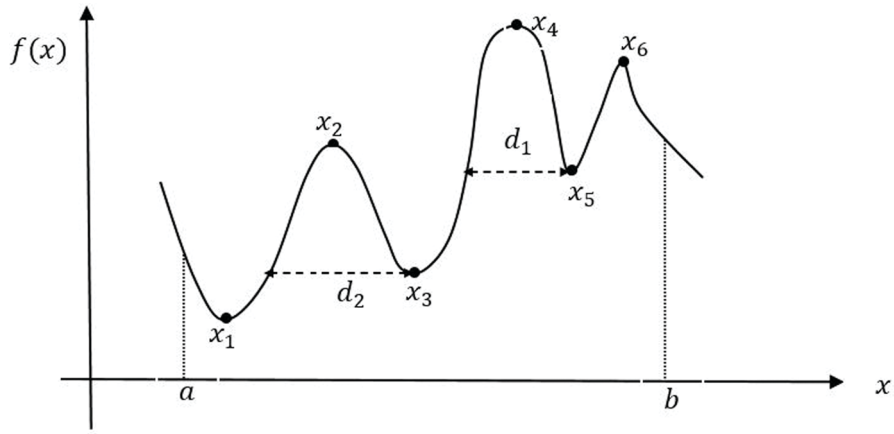


FIGURE 1.11 Local and global solutions; x_1, x_3, x_5 – local minima and x_1 – global minimum x_2, x_4, x_6 – local maxima and x_4 – global maximum.

path exist whose length is at most a specified value, say, L_b?". Brief notes on the computational complexity and categorization of these problems are given in Appendix 1.

1.3.2 Local and Global Solutions

The brute-force technique of Table 1.1 may throw up many local solutions. The global solution is then obtained by choosing the extremum of the available local solutions. A local solution $\boldsymbol{x}^*$ (a maximum or a minimum depending on the problem) is a solution defined with respect to its neighbourhood. A neighbourhood is a region in the vicinity of $\boldsymbol{x}^*$ measured through a 'metric' $d(\boldsymbol{x}, \boldsymbol{y})$. An intuitive understanding of a metric can be had in terms of a 'measure' which, in the present case of TSP, is the Euclidian distance between any two points (cities) $\boldsymbol{x}$ and $\boldsymbol{y}$. The reader can refer to Appendix 1 for additional details on a metric and its properties. One may define an ε-neighbourhood around any $\boldsymbol{x}$ as $\Psi(\boldsymbol{x}) = \{\boldsymbol{y} \mid \boldsymbol{y} \in \mathcal{D} \cap d(\boldsymbol{x}, \boldsymbol{y}) \leq \varepsilon\}$ $\boldsymbol{x}^*$ is a local minimum if $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) \forall \boldsymbol{x} \in \Psi$. In Figure 1.11, we show a 1-D case with $\mathcal{D} = [a, b]$. $f(x)$ has x_3 and x_5 as local minima and x_2 and x_6 as local maxima. Note that x_5 is the local minimum for its neighbourhoods with $\varepsilon \leq d_1$. Similarly, for $\varepsilon \leq d_2$, x_3 is the local minimum for its neighbourhoods. x_1 is the global minimum.

$\boldsymbol{x}^*$ is the global minimum if $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) \forall \boldsymbol{x} \in \mathcal{D}$ and it is the global maximum if $f(\boldsymbol{x}^*) \geq f(\boldsymbol{x}) \forall \boldsymbol{x} \in \mathcal{D}$. In Figure 1.11, x_1 and x_4 are the global minimum and maximum respectively. It is obvious that for a unimodal (strictly convex) function (Figure 1.6), the local and global solutions coincide.

In general, most of the optimization methods may lead to local solutions only. This is particularly so for optimization problems of large dimension where no prior knowledge on the global solution is available. So much so, the endeavour is more often to find a near-optimal solution. Clearly, it is a trade-off between large computational costs in arriving at the global solution through a finely chiselled search and the

practical limitations on such a search that forces us to accept the best available local extremum as an approximation to the global solution.

1.3.3 Solution to TSP by Metropolis Algorithm: The Probabilistic Route[1]

Suppose that we wish to devise a selective search as a more efficient alternative to the brute-force technique. A selective search must be accompanied by a probabilistic criterion to accept or reject a permutation (also called a configuration). This in turn requires the search to be iterative to improve over the solution at the last step. Metropolis algorithm (Metropolis *et al.* 1953) is one such search. The algorithm aims at obtaining approximate solutions to numerical problems where sampling is required from a large but finite set (Diaconis and Saloff-Coste 1998) as in TSP. It has been widely exploited in many scientific fields – statistical mechanics, econometrics, physics and computing science (Beichl and Sullivan 2000, Tanizaki 2004, Gould and Tobochnik 2010). Indeed, it is one of the top ten most used algorithms in the twentieth century (Dongarra and Sullivan 2000). See Bonomi and Lutton (1984) for a more detailed and broad-based review of the algorithm.

Let $\boldsymbol{x}_k$ be an available solution in terms of a Hamiltonian cycle and $f\left(\boldsymbol{x}_k\right)$ the objective function value equal to the corresponding tour length at an iteration k. Suppose that we generate another trial solution $\hat{\boldsymbol{x}}_k$ followed by corresponding $f\left(\hat{\boldsymbol{x}}_k\right)$. The details on how this trial solution is generated will be shortly provided. Now, assuming the optimization problem to be one of minimization, if the change in the objective function value $\Delta f = f\left(\hat{\boldsymbol{x}}_k\right) - f\left(\boldsymbol{x}_k\right) < 0$, the trial solution $\hat{\boldsymbol{x}}_k$ is accepted. Otherwise also (i.e. even if $\Delta f > 0$), the solution is accepted with some probability p_k. This is an interesting part of the Metropolis algorithm. In case one proceeds to the next step only when $\Delta f < 0$, the method may in all prospects get trapped in a local optimum. On the other hand, by the chance-acceptance of an unfavourable solution, a wider exploration of the feasible space for a better local solution (perhaps closer to the global one) is possible. Table 1.4 describes the implementation of the Metropolis algorithm.

TABLE 1.4
Metropolis Algorithm as Applied to TSP – Implementation

(i) Any configuration in the feasible space qualifies as the initial solution $\boldsymbol{x}_0$.

(ii) Suppose that the solution at iteration k is $\boldsymbol{x}_k$. It is required to generate a new trial solution $\hat{\boldsymbol{x}}_k$ for comparison. One way to obtain $\hat{\boldsymbol{x}}_k$ is by swapping the connections of any two cities in $\boldsymbol{x}_k$(see Figure 1.12). Two such cities may be randomly picked from the integer set $\{1,2,\ldots,N\}$. Care must be exercised to ensure that the two differ from each other.

(iii) The next issue is to fix p_k, the acceptance probability (see Appendix 1 for a basic introduction to probability theory). In the initial stages, a high value is set for p_k so that an explorative search in the feasible space is possible without getting trapped in local optima. As iterations progress, p_k is gradually lowered in order to consolidate on the initial gains made and contain the fluctuations around the final solution.

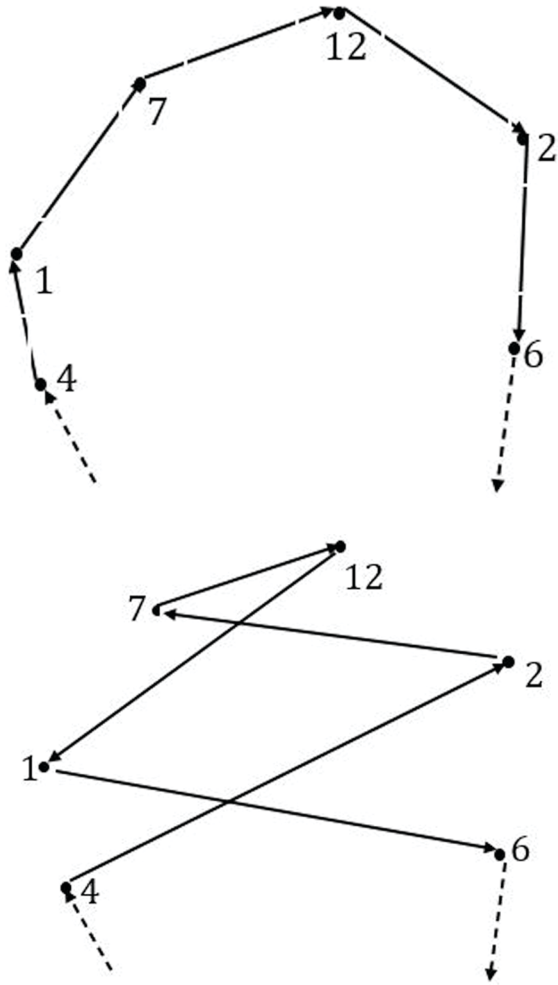


FIGURE 1.12 TSP by Metropolis algorithm; $N = 12$ (only a part of a Hamiltonian cycle is shown in the figure): (a) state x_k, (b) state $\hat{x}_k$ after swapping the connections of cities 1 and 2.

Since the above iterative process involves simulation of random numbers at each iteration (in steps ii and iii), it belongs to a wider class of methods known as Monte Carlo (MC) simulation methods (Janke 2008) (see Appendix 3 for MC simulation). In this respect, step (iii) needs further elaboration. Let us treat $\boldsymbol{x}_k$ as a sample solution or a state in the MC set-up. Metropolis algorithm assumes that $P(\boldsymbol{x})$ representing the probability of being in a state $\boldsymbol{x}$ with tour length $f(\boldsymbol{x})$ is given by the Boltzmann distribution:

$$P(\boldsymbol{x}) = \frac{e^{-\beta f(x)}}{Z_\beta} \tag{1.10}$$

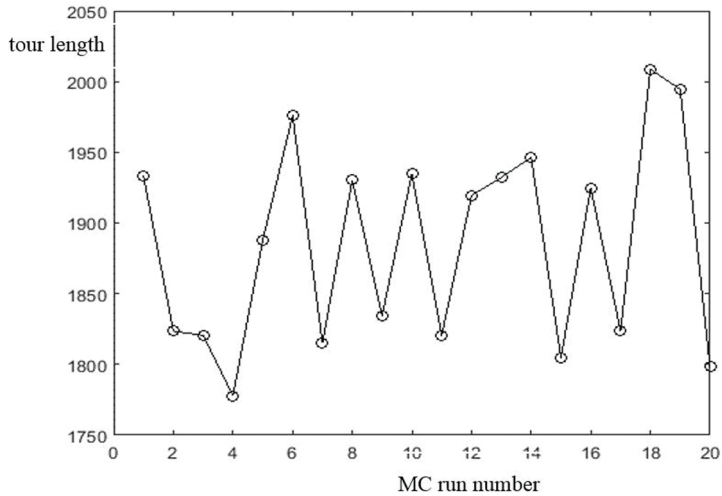


FIGURE 1.13a TSP by selective search based on Metropolis algorithm; $N = 12$, $T_0 = 5000$, $c = 0.99$, solutions from 20 independent MC runs, minimum tour length of 1778 units at the fourth MC run.

where $\beta = \left(K_B T\right)^{-1}$ with K_B denoting the Boltzmann constant (presently assumed to be unity) and T a parameter. Specific to TSP, T pertains to an inverse-length parameter (with the length represented by $f\left(\boldsymbol{x_k}\right)$). $Z_\beta = \sum_{x_k \in \mathcal{D}} e^{-\beta f(x_k)}$ is the normalization constant (partition function in statistical mechanics). Obviously, the normalized $P\left(\boldsymbol{x}\right)$ satisfies the axiom of probability $\sum_{x_k \in \mathcal{D}} P\left(\boldsymbol{x_k}\right) = P\left(\mathcal{D}\right) = 1$. With $P\left(\hat{\boldsymbol{x}}_{\boldsymbol{k}}\right)$ similarly defined for the trial solution $\hat{\boldsymbol{x}}_{\boldsymbol{k}}$, the Metropolis algorithm computes the change $\Delta f_k = f\left(\hat{\boldsymbol{x}}_{\boldsymbol{k}}\right) - f\left(\boldsymbol{x_k}\right)$ and the ratio $\dfrac{P\left(\hat{\boldsymbol{x}}_{\boldsymbol{k}}\right)}{P\left(\boldsymbol{x_k}\right)} = e^{-\beta \Delta f_k}$. The ratio corresponds to the acceptance probability p_k (in step iii above) of transition from the state $\boldsymbol{x_k}$ to the state $\hat{\boldsymbol{x}}_{\boldsymbol{k}}$. Now, we generate a uniformly distributed random number $u \in [0,1]$ and accept the change from $\boldsymbol{x_k}$ to $\hat{\boldsymbol{x}}_{\boldsymbol{k}}$ when $\Delta f \le 0$ or if $u \le p_k$ when $\Delta f > 0$. Otherwise, we reject the change, retain $\boldsymbol{x_k}$ and proceed to the next iteration. The one last issue in step (iii) is the schedule of reducing the parameter T contained in β. Initially we keep $T = T_0 > 0$ high so that p_k is high and then reduce it with progressing iterations according as $T_k = cT_0$ where $0 < c < 1$.

Figures 1.13a–c show the search result for $N = 12$ with the distance matrix given in Table 1.2.

Figure 1.13a shows the results from 20 independent MC runs, each spanning 1000 iterations. The fourth MC run corresponds to the global minimum whose Hamiltonian

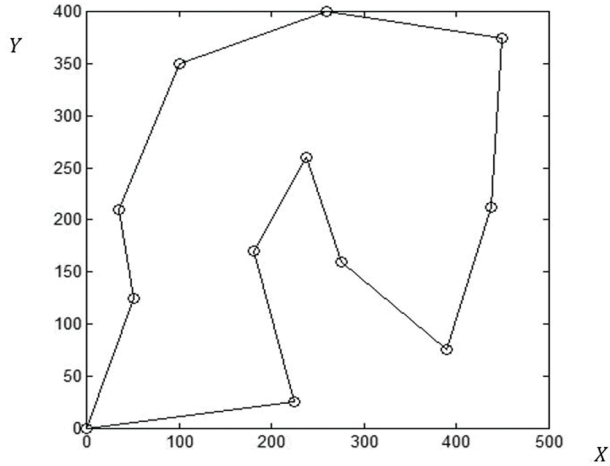


FIGURE 1.13b TSP by selective search based on Metropolis algorithm; $N = 12$, $T_0 = 5000$, $c = 0.99$, solution from the fourth MC run (Hamiltonian cycle of minimum length = 1778 units) – see Figure 1.13a.

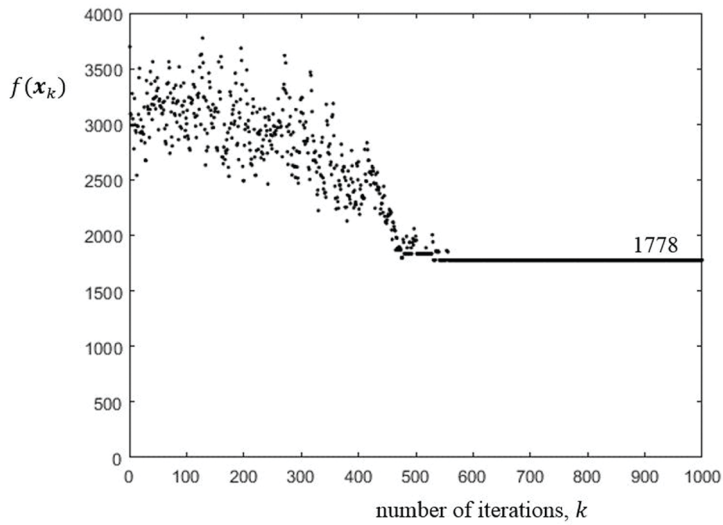


FIGURE 1.13c TSP by selective search based on Metropolis algorithm; $N = 12$, $T_0 = 5000$, $c = 0.99$, evolution of solution (for the fourth MC run) vs. iteration number; minimum tour length = 1778 units, total execution time for 20 MC runs = 77.735 s.

cycle (tour) is shown in Figure 1.13b. The result is the same as obtained by the brute-force technique (see Figure 1.10). Convergence of the result from this fourth MC run is shown in Figure 1.13c. It is significant to note that the selective search method has hugely reduced the computational time – which is now a small fraction of the time taken by the brute-force technique (see Table 1.3 for $N = 12$). The results on the execution times are with reference to a laptop version of I7 with 8 GB RAM. For larger-sized problems, more independent MC simulations would be required to hit an acceptable solution. Yet, the method is found to be fast and efficient. This is highlighted by an example with large $N = 50$ cities. It is computationally an impossible task (Fogel 1988) to solve this case using the brute-force technique. The cities are randomly located in a 500×500 square of appropriate units. The results from three independent MC runs are shown in Figure 1.14.

In an interesting application to statistical physics (Landau and Binder 2000), the Metropolis algorithm is used to study phase transitions in magnetic materials with temperature as a parameter. In this study, $f(\boldsymbol{x})$ stands for the energy of a (magnetic) spin configuration $\boldsymbol{x}$ and $P(\boldsymbol{x})$ for the probability of being in a thermal equilibrium state, at a given temperature, with the configuration $\boldsymbol{x}$. The parameter T represents temperature and the objective is to find the state of minimum energy at each T. We will discuss this problem in Chapter 3 in connection with evolutionary optimization methods and specifically the simulated annealing technique.

1.4 THE BRACHISTOCHRONE PROBLEM

We are finally back to continuous optimization and hence the brachistochrone problem shown in Figure 1.5, which is redrawn with some more details in Figure 1.15. Let v be the speed of the bead of mass m, at any point (x, y) on the curve $y(x)$. The kinetic and potential energies are respectively given by $\frac{1}{2}mv^2$ and mgy. Hence, $v = \sqrt{2gy}$ where 'g' is the acceleration due to gravity. The distance traversed by the bead along the curve in differential time dt is:

$$ds = \sqrt{dx^2 + dy^2} = dx\sqrt{1 + \left(\frac{dy}{dx}\right)^2} \text{ so that } dt = \frac{ds}{v} \tag{1.11}$$

The brachistochrone problem is thus an optimization problem of finding the minimum time $T(y)$ with respect to the path $y(x)$, given by:

$$0 < T(y) = \int_a^b dt = \int_a^b \frac{ds}{v} = \int_a^b \sqrt{\frac{1 + \left(\frac{dy}{dx}\right)^2}{2gy}}\, dx \tag{1.12}$$

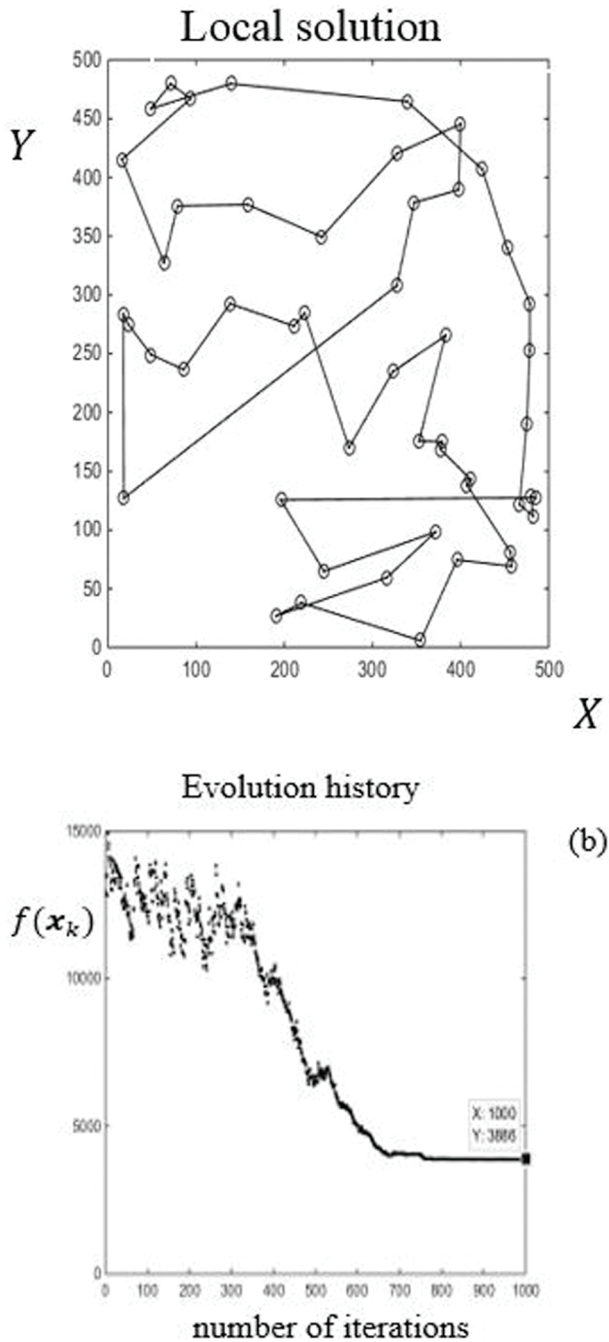


FIGURE 1.14a–b TSP with $N = 50$ cities; local solutions and evolution histories: (a) and (b) first MC run, tour length = 3886 units and execution time = 25.81 s.

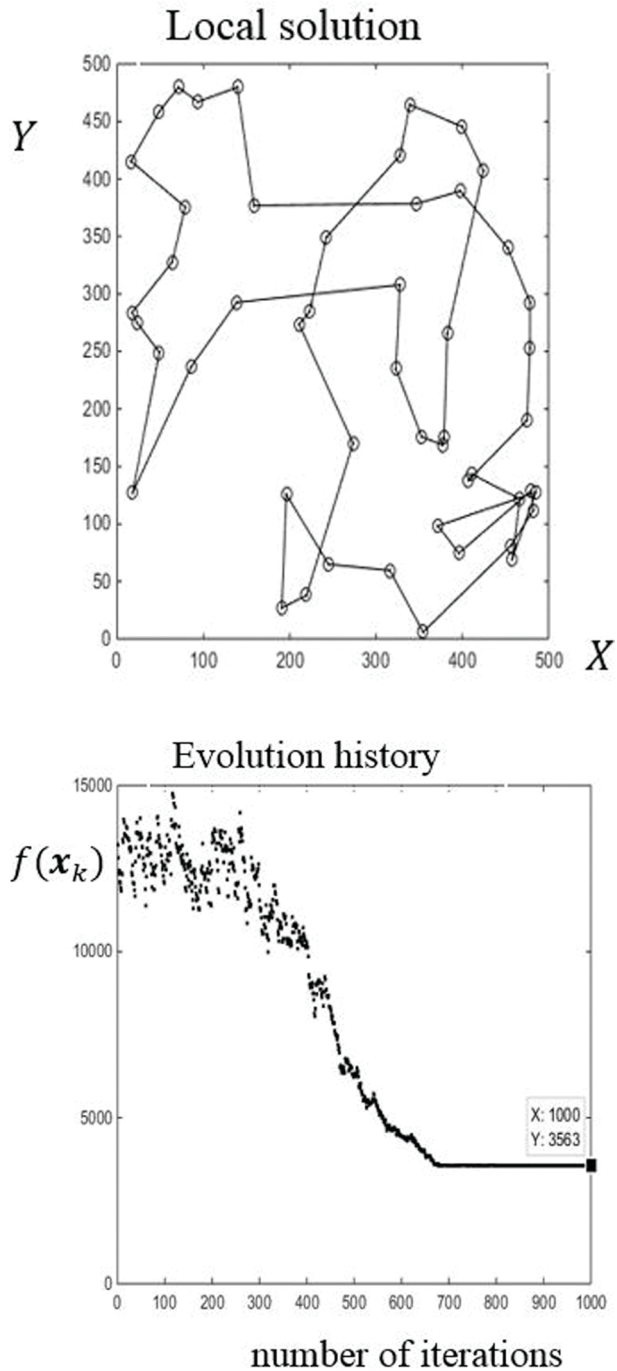


FIGURE 1.14c–d TSP with $N = 50$ cities; local solutions and evolution histories: (c) and (d) second MC run, tour length = 3563 units and execution time = 25.40 s.

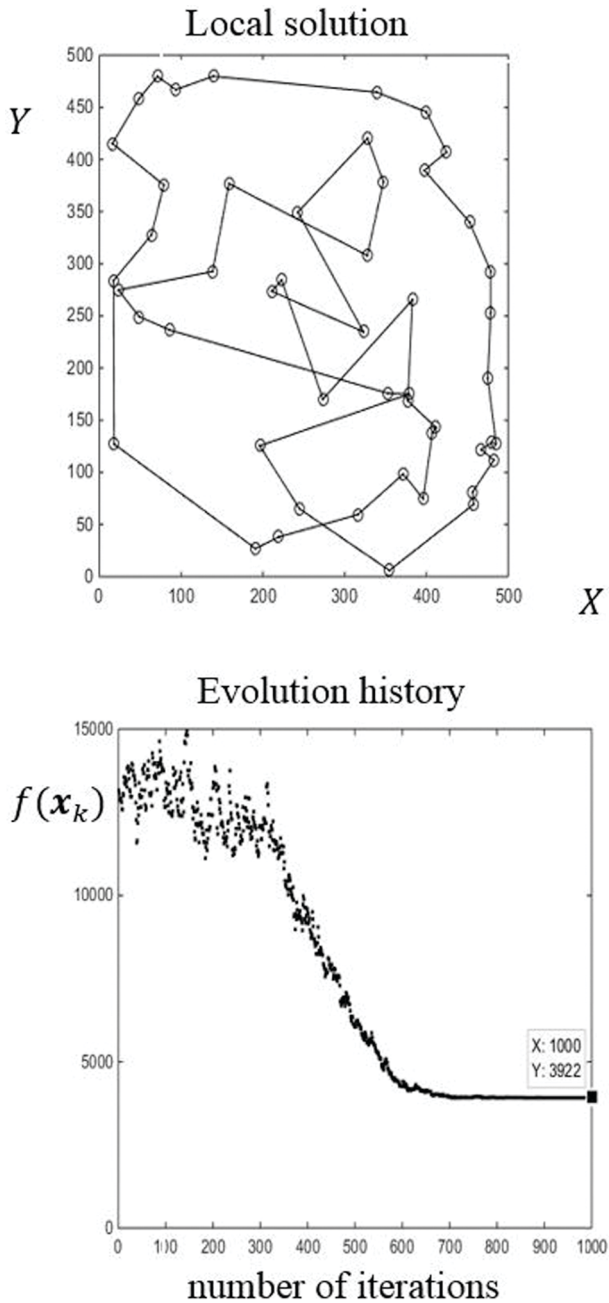


FIGURE 1.14e–f TSP with $N = 50$ cities; local solutions and evolution histories: (e) and (f) third MC run, tour length = 3922 units and execution time = 27.03 s.

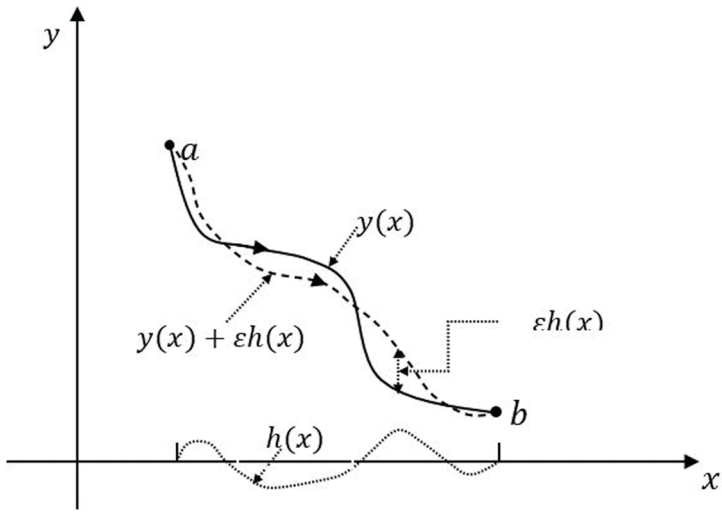


FIGURE 1.15 A brachistochrone problem; dark line – a typical path $y(x)$ between the fixed points a and b, dashed line – a varied path $y(x)+\varepsilon h(x)$, $\varepsilon \in \mathbb{R}$.

1.4.1 Solution of the Brachistochrone Problem by Variational Approach

With $y' = \dfrac{dy}{dx}$, T in Equation (1.12) is regarded as a functional of y and y' – a real-valued function involving other functions as arguments. In general, while writing a functional $I(y) = \int_a^b L(y, y', x)\,dx$, where L is a function of y, y' and x, the objective is to find its extremum. This is a problem of functional optimization. Similar to the extrema of a function, at which the first derivatives vanish (see the Introduction to this chapter for the optimality conditions), the extrema of a functional may also be obtained by finding the argument functions for which the functional derivative or the first variation δI vanishes. Let $y(x)$ be the function, unknown at this stage, that extremizes $I(y)$. Consider a possible varied path $y_\epsilon(x) = y(x)+\varepsilon h(x)$ as shown in Figure 1.15 where ε is a real constant. Let $h(x)$ be an arbitrary function satisfying the boundary conditions (BCs) $h(a) = h(b) = 0$. Thus,

$$I(y_\epsilon) = \int_a^b L(y(x)+\varepsilon h(x), y'(x)+\varepsilon h-(x), x)\,dx \tag{1.13}$$

Now, $I(y_\epsilon)$ may be considered to be a function of ε alone. At the extremum, the first variation δI to be zero is equivalent to the first derivative $\left.\dfrac{dI(\varepsilon)}{d\varepsilon}\right|_{\varepsilon=0}$ being zero. $\dfrac{dI(\varepsilon)}{d\varepsilon}$ is obtained as:

$$\frac{dI(\varepsilon)}{d\varepsilon} = \frac{d}{d\varepsilon}\int_a^b L\left(y(x)+\varepsilon h(x), y'(x)+\varepsilon h'(x), x\right)dx$$

$$= \int_a^b \left(\frac{\partial L}{\partial x}\frac{dx}{d\varepsilon} + \frac{\partial L}{\partial y_\epsilon}\frac{dy_\epsilon}{d\varepsilon} + \frac{\partial L}{\partial y'_\varepsilon}\frac{dy'_\varepsilon}{d\varepsilon}\right)dx$$

$$= \int_a^b \left(\frac{\partial L}{\partial y_\epsilon}h + \frac{\partial L}{\partial y'_\varepsilon}h'\right)dx \left(\text{since}\frac{dx}{d\varepsilon}=0, \frac{dy_\varepsilon}{d\varepsilon}=h, \frac{dy_\varepsilon}{d\varepsilon}=h'\right)$$

$$= \int_a^b \frac{\partial L}{\partial y_\epsilon}h dx + \int_a^b \frac{\partial L}{\partial y'_\varepsilon}h' dx \tag{1.14}$$

Integrating by parts the second term on the RHS of the last equality in Equation (1.14), one gets:

$$\frac{dI(\varepsilon)}{d\varepsilon} = \int_a^b h\frac{\partial L}{\partial y_\epsilon}dx - \int_a^b h\frac{d}{dx}\left(\frac{\partial L}{\partial y'_\varepsilon}\right)dx + \frac{\partial L}{\partial y'_\varepsilon}h\Bigg|_a^b \tag{1.15}$$

Since $h(x)$ vanishes at the endpoints a and b, one has:

$$\frac{dI(\varepsilon)}{d\varepsilon} = \int_a^b h\frac{\partial L}{\partial y_\epsilon}dx - \int_a^b h\frac{d}{dx}\left(\frac{\partial L}{\partial y'_\varepsilon}\right)dx$$

$$= \int_a^b h(x)\left\{\frac{\partial L}{\partial y_\epsilon} - \frac{d}{dx}\left(\frac{\partial L}{\partial y'_\varepsilon}\right)\right\}dx \tag{1.16}$$

Now imposing the optimality condition $\left.\frac{dI(\varepsilon)}{d\varepsilon}\right|_{\varepsilon=0} = 0$ for the extremum of $I(\varepsilon)$, we obtain:

$$\int_a^b h(x)\left\{\frac{\partial L}{\partial y} - \frac{d}{dx}\left(\frac{\partial L}{\partial y'}\right)\right\}dx = 0 \tag{1.17}$$

Since $h(x)$ is an arbitrary function, the quantity within the curly brackets must be zero. So $y(x)$ extremizes the functional $I(y)$ under the condition:

$$\frac{\partial L}{\partial y} - \frac{d}{dx}\left(\frac{\partial L}{\partial y'}\right) = 0 \tag{1.18}$$

Equation (1.18) is known as the Euler-Lagrange (EL) equation to solve for the functions $y(x)$ and $y'(x)$ that render I stationary. In classical mechanics, L is known as the Lagrangian. In the brachistochrone problem, the action integral in Equation (1.12) has no explicit dependence on x. With $L = \sqrt{\frac{1+y'^2}{2gy}}$, one has:

$$\frac{\partial L}{\partial y} = -\frac{1}{2\sqrt{2gy}}\frac{1+y'^2}{y}$$

$$\frac{d}{dx}\left(\frac{\partial L}{\partial y'}\right) = \frac{1}{\sqrt{2gy\left(1+y'^2\right)}}\left(\frac{y''}{1+y'^2} - \frac{1}{2}\frac{y'^2}{y}\right) \tag{1.19}$$

Here $y'' := \frac{d^2y}{dx^2}$. Substitution of the above in EL equation (1.18) and simplification lead to an ordinary differential equation (ODE):

$$2yy'' + 1 + y'^2 = 0$$

$$\Rightarrow y'\left(2yy'' + 1 + y'^2\right) = 0$$

$$\Rightarrow \frac{d}{dx}\left(y + yy'^2\right) = 0$$

$$\Rightarrow y\left(1+y'^2\right) = A \in \mathbb{R}\ \left(\text{by integrating with respect to } x \text{ in the last step}\right) \tag{1.20}$$

From the last equation, one may have:

$$y' = \sqrt{\frac{A-y}{y}} \tag{1.21}$$

While the ODE above is solvable by the separation-of-variable method, it is more intuitive if a solution is sought in terms of the parameter θ, the angle made by the tangent to the curve with the vertical (Figure 1.16).

With $\frac{dy}{dx} = \cot\theta = \sqrt{\frac{A-y}{y}}$ (from Equation 1.21), one gets:

$$y = \frac{A}{2}\left(1 - \cos 2\theta\right) \tag{1.22a}$$

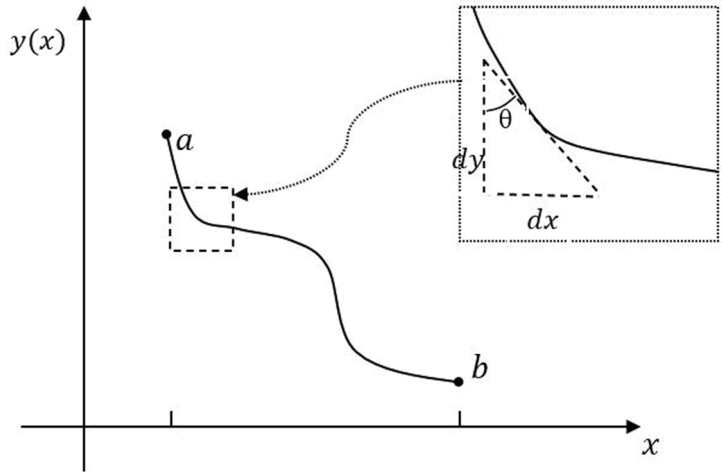


FIGURE 1.16 The brachistochrone problem (also refer to Figure 1.15).

It follows that:

$$\frac{dx}{d\theta} = \frac{dx}{dy}\frac{dy}{d\theta} = A\sin 2\theta\tan\theta = A\left(1-\cos 2\theta\right) \Rightarrow x = \frac{A}{2}\left(2\theta - \sin 2\theta\right) \quad (1.22b)$$

The parametric equations (1.22) give the solution $y(x)$ for the brachistochrone problem. These equations correspond to an inverted cycloid, an evolute[2] of a circle with radius $A/2$. Knowing the coordinates of a and b, the constant A may be obtained from Equations (1.22). The minimum time of descent from a to b along the path given by Equations (1.12) is now obtained as:

$$T\left(y(x)\right) = \int_a^b \sqrt{\frac{1+\left(\frac{dy}{dx}\right)^2}{2gy}}\,dx$$

$$= \sqrt{\frac{2A}{g}}\int_{\theta_a}^{\theta_b} d\theta = \sqrt{\frac{2A}{g}}\left(\theta_b - \theta_a\right) \quad (1.23)$$

where θ_a and θ_b are the angles to the vertical at points a and b. Suppose that a and b are the starting and lowest points of the cycloid. Then θ_a and θ_b are 0 and $\frac{\pi}{2}$ respectively and the minimum time of descent is $\pi\sqrt{\frac{A}{2g}}$. In case the bead is allowed to traverse along a straight line joining these specific a and b, the time of descent may be directly obtained by analogy with a freely falling body with acceleration $g\cos\theta$ along

the line. It is equal to $\sqrt{4+\pi^2}\sqrt{\frac{A}{2g}}$ which is approximately 18.6% more than the time taken for the cycloid.

Bernoulli's original solution to the brachistochrone problem

Bernoulli (1697) solved the brachistochrone problem via Fermat's principle of least time which is equivalent to Snell's law of refraction in optics. The proof is interesting in its own right as it offers a motivating glimpse into a great mind. The principle states that a light ray takes a path, between two points, that minimizes the travelling time between the two points. The speed of light increases as it enters a medium of lower optical density i.e., of decreasing index of refraction. If v is the speed of light in a medium, it is inversely related to its refractive index ϑ by the relationship $v = c / \vartheta$ where c is the speed of light in vacuum. By Snell's law of refraction (Figure 1.17):

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{\mu_{rarer}}{\mu_{denser}} = \frac{v_1}{v_2} \tag{1.24}$$

v_1 and v_2 are the speed of light in the two media and μ_{denser}, μ_{rarer} are their respective refractive indices.

Fermat's principle of least time follows from Figure 1.17. The time of travel by light from point A to B is:

$$T(x) = \frac{\sqrt{x^2+h_1^2}}{v_1} + \frac{\sqrt{(s-x)^2+h_2^2}}{v_2} \tag{1.25}$$

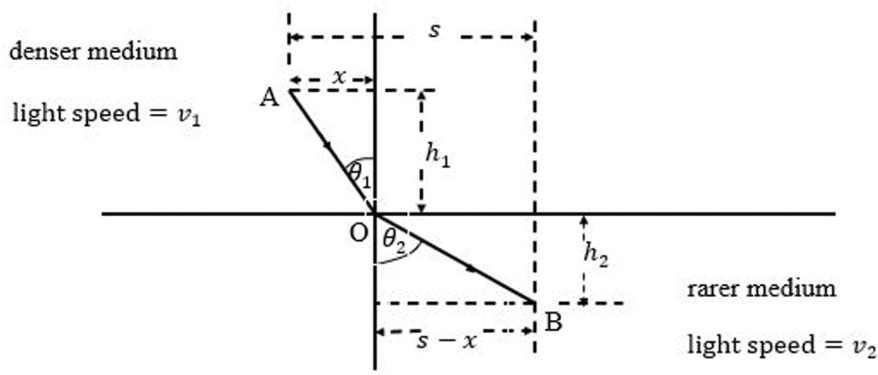


FIGURE 1.17 Fermat's principle of least time; v_1 and v_2 – speed of light in the two media, AO – incident ray, OB– refracted ray.

For the time to be minimum, the necessary condition is $\frac{dT}{dx} = 0$ and thus:

$$\frac{dT}{dx} = \frac{2x}{\sqrt{x^2 + h_1^2}} \frac{1}{v_1} - \frac{2(s-x)}{\sqrt{(s-x)^2 + h_2^2}} \frac{1}{v_2} = 0$$

$$\Rightarrow \frac{\sin\theta_1}{v_1} = \frac{\sin\theta_2}{v_2} \tag{1.26}$$

which is the same as Snell's law of refraction (Equation 1.24). The significance of the result is that as light travels through multi-layered media with each medium rarer than the previous one, it always takes the least time along the path consisting of the incident and refracted rays and satisfying the condition:

$$\frac{\sin\theta_1}{v_1} = \frac{\sin\theta_2}{v_2} = \frac{\sin\theta_3}{v_3} = \ldots = constant \tag{1.27}$$

As the number of layers increases to infinity and the thickness of each medium decreases to zero, the path of light tends to a smooth curve satisfying the following condition at each point:

$$\frac{\sin\theta}{v} = constant \tag{1.28}$$

Bernoulli applied this condition to the brachistochrone problem and arrived at the optimum path $y(x)$. Refer to Figure 1.18 where Bernoulli's diagram for the brachistochrone problem is redrawn from Struik (1986).

As the bead traverses along ABM through the depth AC, the curve AH represents the increase in velocity of the bead. This is analogous to the increase in speed of light as it goes through the infinitely layered medium with decreasing μ. If v is the velocity attained by the bead as it reaches the point M, it is given by CH. With AC = x, the velocity $v = \text{CH} = \sqrt{2gx}$. Utilizing the infinitesimal triangle Mmn in Figure 1.18 to write $\sin\theta = \frac{nm}{Mm} = \frac{dy}{\sqrt{dx^2 + dy^2}}$, and imposing the condition in Equation (1.28), one gets:

$$\frac{\frac{dy}{\sqrt{dx^2 + dy^2}}}{v} = C \tag{1.29}$$

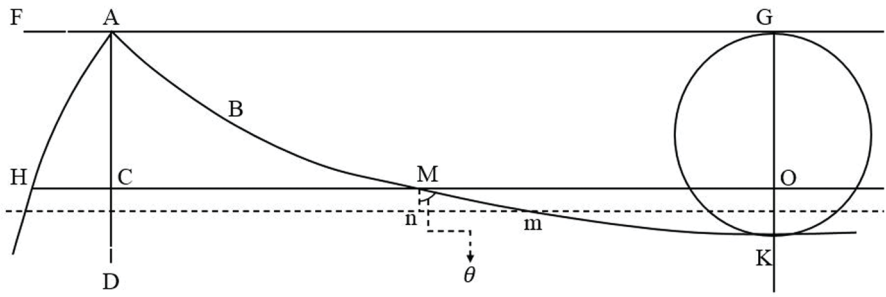


FIGURE 1.18 Bernoulli's diagram for the brachistochrone problem adapted from Struik [1986] – an optical analogy: ABMK – the least time path and is the brachistochrone solution, point A – start of luminous light, AH – representation of the increasing velocity of the particle during its descent along ABMK, CM – horizontal coordinate y, AC – vertical coordinate x, CH – velocity v, nm = dy, Mn = dx.

Here C is a real constant. We rewrite Equation (1.29) as:

$$\frac{\dfrac{dy}{dx}}{\sqrt{1+\left(\dfrac{dy}{dx}\right)^2}} = C\sqrt{2gx}$$

$$\Rightarrow \frac{dx}{dy} = \sqrt{\frac{1-2gC^2x}{2gC^2x}} = \sqrt{\frac{A-x}{x}} \tag{1.30}$$

$A = \dfrac{1}{2gC^2}$ is a real constant. Noting the reversal in the notation for x and y axes in Figure 1.18 vis-à-vis Figure 1.16, $\dfrac{dx}{dy}$ in Equation (1.30) is the slope of the optimal path for the brachistochrone problem. This is the same as the one in Equation (1.21) earlier obtained by the variational calculus approach. The above proof of Bernoulli is a demonstration of two phenomena of entirely two separate fields of physics – one from optics and the other from mechanics – exhibiting the same character (as claimed by Bernoulli himself).

1.5 MORE ON FUNCTIONAL OPTIMIZATION: HAMILTON'S PRINCIPLE

Having posed and solved the brachistochrone problem as one in functional optimization, we now discuss another related approach: Hamilton's principle – a method of great relevance in classical mechanics. One finds in Hamilton's principle (Meirovitch 1970, Goldstein *et al.* 1980) a generalization to higher-order systems using the

stationarity (see Section 1.3) of a functional. Following the methods of variational calculus, this principle establishes the EL equations governing the system dynamics. These are partial differential equations (PDEs) for continuous systems wherein the parameters – mass, stiffness and damping – appear as specified continuous functions. In the case of discrete systems, either inherently discrete or lumped parameter systems, the EL equations are ODEs. For dynamical systems of relatively simple description, Newton's laws of physics may help in directly obtaining the governing differential equations (DEs) of motion. Hamilton's principle is of an integral form in terms of scalar quantities such as 'work' and 'energy' in contrast to the vector quantities in Newtonian force balance. It is more general, to the extent that it even provides the basis for the formulation of the FEM as a means to discretization and solution of the equations of motion in continuous systems with complex geometry. For a conservative system,[3] the action integral or the functional I is:

$$I = \int_{t_1}^{t_2} L dt \tag{1.31}$$

where the Lagrangian $L = \mathrm{T} - \mathrm{V}$; T and V, respectively, denote the system kinetic and potential energies. Equation (1.31) is similar to the action integral in Equation (1.13) of the brachistochrone problem. By Hamilton's principle, we have the necessary condition for stationarity of I as:

$$\delta I = \delta \int_{t_1}^{t_2} L dt = 0 \tag{1.32}$$

For systems acted upon by external loads and having dissipation, the stationarity condition is given by the extended Hamilton's principle:

$$\delta I = \delta \int_{t_1}^{t_2} \left(L + W_{nc} \right) dt = 0 \tag{1.33}$$

W_{nc} is the work done by the non-conservative forces. A force is said to be non-conservative, if the work done by the force in moving an object from an initial position to a final position is dependent on its path – not just the two boundary points joined by the path.

Example 1.1. Application of functional optimization in deriving the equations of motion for a continuous system

For continuous systems in mechanics such as 1-D rods and beams, 2-D plates and 3-D solid structures, the Lagrangian $L = \mathrm{T} - \mathrm{V}$ is usually in the form of an integral over its domain. The integrand here is a function of spatial coordinates $\boldsymbol{x} := \left(x, y, z \right)^T$ and its derivatives with respect to $\boldsymbol{x}$ in addition to derivatives with respect to time t. The integrand is called the Lagrangian density. As an example, consider an axially vibrating rod shown in Figure 1.19. Let us derive the EL equations by the variational approach.

$f_A \left(x, t \right)$ = axial force density per unit length

Solution. If the axial displacement field variable is denoted by $y(x,t)$, the kinetic energy $\mathrm{T} = \frac{1}{2}\int_0^l m\dot{y}^2 dx$ and the potential energy $\mathrm{V} = \frac{1}{2}\int_0^l EAy'^2 dx$ where $\dot{y} = \frac{dy}{dt}$ and $y' = \frac{dy}{dx}$. Thus $L = \frac{1}{2}\int_0^l \left(m\dot{y}^2 - EAy'^2\right)dx$ and $\mathrm{L} = \frac{1}{2}\left(m\dot{y}^2 - EAy'^2\right)$, the latter being the Lagrangian density. Here the system parameters E, A and m are considered functions of the spatial variable x. Note that the differential $d\mathrm{L}$ is exact. If the system is acted upon by an external force density $f_A(x,t)$ distributed over its length, $W_{nc} = \int_0^l f_A(x,t)\, y(x,t)dx$ – the integrand here cannot typically be obtained as the gradient of a smooth/analytic function. The action integral I is:

$$I = \int_{t_1}^{t_2} \left(L\left(\dot{y}, y'\right) + W_{nc}\right)dt \tag{1.34}$$

Now, we aim at finding the path $y(x,t)$ that extremizes I. By the extended Hamilton's principle, the necessary condition as in Equation (1.33) takes the form:

$$\delta I = 0 = \int_{t_1}^{t_2}\int_0^l \left(\frac{\partial \mathrm{L}}{\partial \dot{y}}\delta\dot{y} + \frac{\partial \mathrm{L}}{\partial y'}\delta y' + f_A\delta y\right)dxdt \tag{1.35}$$

$\delta y(x,t)$ is the virtual displacement over the true path satisfying $\delta y(x,t_1) = 0 = \delta y(x,t_2)$. It follows that $\delta\dot{y}(x,t)$ is the virtual velocity and $\delta y'$, the first-order partial derivative with respect to x. To express all variations in terms of δy, we integrate by parts the first two terms in the parenthesis on the extreme RHS of Equation (1.35) with respect to t and x, respectively. With the integration defined over finite spatial domain and time interval, the operator pairs $(\delta, \frac{\partial}{\partial t})$ and $(\delta, \frac{\partial}{\partial x})$ commute. Similarly, integrations with respect to t and x are also interchangeable.
Then we have:

$$\int_{t_1}^{t_2} \frac{\partial \mathrm{L}}{\partial \dot{y}}\delta\dot{y}\, dt = \int_{t_1}^{t_2} \frac{\partial \mathrm{L}}{\partial \dot{y}}\delta\left(\frac{\partial y}{\partial t}\right)dt$$

$$= \int_{t_1}^{t_2} \frac{\partial \mathrm{L}}{\partial \dot{y}}\frac{\partial(\delta y)}{\partial t}dt \text{ (by commutativity of } \delta \text{ and } \frac{\partial}{\partial t})$$

$$= \frac{\partial \mathrm{L}}{\partial \dot{y}}\delta y(x,t)\Big|_{t_1}^{t_2} - \int_{t_1}^{t_2} \frac{\partial}{\partial t}\left(\frac{\partial \mathrm{L}}{\partial \dot{y}}\right)\delta y\, dt = -\int_{t_1}^{t_2} \frac{\partial}{\partial t}\left(\frac{\partial \mathrm{L}}{\partial \dot{y}}\right)\delta y\, dt \tag{1.36a}$$

(since $\delta y = 0$ at t_1 and t_2)

$$\int_0^l \frac{\partial \mathrm{L}}{\partial y'}\delta y' dx = \int_0^l \frac{\partial \mathrm{L}}{\partial y'}\frac{\partial(\delta y)}{\partial x}dx \text{ (by commutativity of } \delta \text{ and } \frac{\partial}{\partial x})$$

$$= \frac{\partial \mathrm{L}}{\partial y'}\delta y(x,t)\Big|_0^l - \int_0^l \frac{\partial}{\partial x}\left(\frac{\partial \mathrm{L}}{\partial y'}\right)\delta y dx \tag{1.36b}$$

Substitution of these expressions in Equation (1.35) gives:

$$\delta I = \int_{t_1}^{t_2} \left[\int_0^l \left\{ -\frac{\partial}{\partial t}\left(\frac{\partial \iota}{\partial \dot{y}}\right) - \frac{\partial}{\partial x}\left(\frac{\partial \iota}{\partial y'}\right) + f_A \right\} \delta y dx + \frac{\partial \iota}{\partial y'} \delta y \Big|_0^l \right] dt = 0 \tag{1.37}$$

Since δy is arbitrary, the condition $\delta I = 0$ is satisfied for all δy if and only if:

$$-\frac{\partial}{\partial t}\left(\frac{\partial \iota}{\partial \dot{y}}\right) - \frac{\partial}{\partial x}\left(\frac{\partial \iota}{\partial y'}\right) + f_A = 0, x \in [0, l]$$

$$\Rightarrow \frac{\partial}{\partial t}\left(\frac{\partial \iota}{\partial \dot{y}}\right) + \frac{\partial}{\partial x}\left(\frac{\partial \iota}{\partial y'}\right) = f_A \tag{1.38}$$

and

$$\left\{\frac{\partial \iota}{\partial y'}\right\} \delta y \Big|_0^l = 0 \tag{1.39}$$

Equation (1.38) is the EL equation which is a PDE in the unknown field $y(x,t)$. Equation (1.39) reveals the possible BCs for the system. Thus, the variational formulation results in the governing PDEs together with the possible sets of BCs. With the specific Lagrangian density ι of the vibrating rod substituted in Equation (1.38), one gets the governing PDE:

$$-\frac{\partial}{\partial x}\left(EA\frac{\partial y}{\partial x}\right) + m\frac{\partial^2 y}{\partial t^2} = f_A \tag{1.40}$$

■

For the rod shown in Figure 1.19, Equation (1.39) is satisfied by the two BCs: $\frac{\partial \iota}{\partial y'} = EA\frac{\partial y}{\partial x} = 0$ at $x = l$ and $y(0,t) = 0$. While $EA\frac{\partial y(l,t)}{\partial x} = 0$ is the natural (force) BC, $y(0,t) = 0$ is known as the essential (geometric) BC. The essential BCs are also known as Dirichlet BCs where the dependent variable [here $y(x,t)$] is directly specified on the boundary of the domain. The natural BCs are of Neumann type where normal derivatives are prescribed on the boundary. To arrive at an explicit solution $y(x,t)$, one needs to solve the PDE (1.40) subject to the prescribed BCs and the

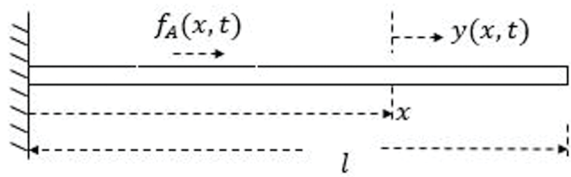


FIGURE 1.19 A continuous system – an axially vibrating rod of length l; $m(x)$ = mass density per unit length, $E(x)$ = Young's modulus of elasticity, $A(x)$ = area of cross-section of the rod.

initial conditions (ICs). The ICs are typically described in the form: $y(x,0)=\alpha(x)$ and $\dot{y}(x,0)=\beta(x)$.

1.5.1 Functional Optimization and Numerical Schemes

The PDE (1.40) is often referred to as the strong form of the governing equation. While the solution to this PDE solves the functional optimization problem, it is generally difficult to arrive at an analytical solution. This clearly highlights the important role that numerical schemes can play in this regard. First note that the solution to the EL Equation (1.40) must be $C^2(0,l)$ with respect to x and $C^2(0,\infty)$ with respect to t. $C^m(a,b)$ is the standard notation used to denote the set of all functions that, together with their first m derivatives, are continuous in (a,b). One may ask if the problem may be posed in such a way that the continuity and smoothness requirements on $y(x,t)$ could be relaxed; this would be helpful in devising a numerical scheme as well. Indeed, this is the underlying philosophy of numerical methods such as Rayleigh-Ritz and weighted residuals (Meirovitch 1967, Finlayson 1972, Roy and Rao 2017). Galerkin, least squares and collocation methods belong to the category of weighted residuals. The Rayleigh-Ritz method involves a functional discretization as an approximation $\tilde{y}(\boldsymbol{x},t)$ to the unknown field variable $y(\boldsymbol{x},t)$:

$$\tilde{y}(\boldsymbol{x},t)=\sum_j Y_j(\boldsymbol{x})q_j(t),\ \ j=1,2,\ldots \tag{1.41}$$

The above is a linear combination of basis or trial functions $Y_j(\boldsymbol{x})$; note that these are known functions. $q_j(t)$ may be treated as generalized coordinates (in that they may not possess any physical attribute). The trial functions are required to be admissible, i.e. they should be continuous, linearly independent and complete (see Appendix 1 for definitions of linear independence and completeness). They need only to satisfy the essential boundary conditions (BCs), but not the natural BCs, since this requirement is already provided for in the variational formulation. Substitution of the assumed solution (1.41) in the action integral I of Equation (1.34) and application of the stationarity condition $\delta I=0$ result in a set of ODEs in unknown functions $q_j(t)$ and solution of these equations by an integration scheme yields the generalized coordinates and thence $y(\boldsymbol{x},t)$ (Clough and Penzien 1982).

Weighted residual methods also use functional discretization as in Equation (1.41). A direct substitution of $\tilde{y}(\boldsymbol{x},t)$ in the system PDE – linear or nonlinear – yields a residual $\mathcal{R}(\tilde{y})$. For example, this residual for the PDE in Equation (1.40) is: $\mathcal{R}(\tilde{y})=\mathcal{A}(\tilde{y})-f_A$ where $\mathcal{A}$ is the differential operator $-\frac{\partial}{\partial x}\left(EA\frac{\partial}{\partial x}\right)+m\frac{\partial^2}{\partial t^2}$. The basic idea is to render the residual a minimum, in some sense, by using a set of orthogonality conditions. In particular, the residual is orthogonalized with respect to certain weight or test functions $U_j,\ j=1,2,\ldots$ such that:

$$<U_{j,}\mathcal{R}> = 0 \Rightarrow \int_{\upsilon} U_j(x)\left\{\mathcal{A}(\tilde{y}) - f_A\right\} d\boldsymbol{x} = 0,\ \ j = 1,2,\ldots \tag{1.42}$$

with Y_j and $U_j \in \mathcal{H}$, the Hilbert space (Appendix 1). $<.,.>$ stands for the inner product.[4] All finite dimensional inner product spaces belong to $\mathcal{H}$.

Equation (1.42) amounts to an orthogonal projection of the residual on a reduced (and finite) dimensional linear (vector) space spanned by the elements of $\boldsymbol{U}(\boldsymbol{x}) = \left\{U_j(\boldsymbol{x}), j = 1,2,\ldots,N\right\}$. This stems from the classical projection theorem – "given a Hilbert space $\mathcal{H}$ and a closed non-empty subspace V of $\mathcal{H}$ and with $u \in \mathcal{H}$, there exists a unique $v \in V$ such that the norm[5] $\|u - v\| \leq \|u - z\|$ for all $z \in V$". Consider the case wherein the trial and test functions belong to the same finite dimensional vector space.

The weighted residual method seeks to find the best approximant within this space in the sense as envisaged through the projection theorem. Intuitively, this is implicit from the fact that a position vector $\boldsymbol{U}$ in $\mathbb{R}^3$ is represented by $\tilde{\boldsymbol{U}}$ in the reduced $\mathbb{R}^2$ with minimum error when the error (residual) $\boldsymbol{U} - \tilde{\boldsymbol{U}}$ is perpendicular (orthogonal) to the $X_1 - X_2$ plane as illustrated in Figure 1.20.

From Equation (1.42), one observes that there are dissimilar restrictions on the smoothness requirements of the trial and test functions. To be specific, for the example PDE (1.40), second order derivatives of $\tilde{y}$ appear in the weighted residual, but not derivatives of U_j. Y_j, $j = 1,2,\ldots$ are required to satisfy both the essential and natural BCs since Equation (1.35) does not contain these conditions. By applying Green's identity (Appendix 1) to the inner product in Equation (1.42), the boundary value problem (BVP) in Equation (1.40) may be transformed to a so-called 'weak

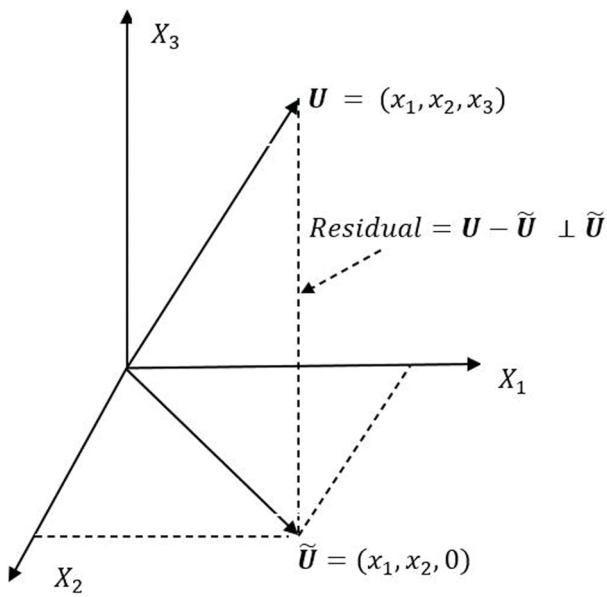


FIGURE 1.20 Orthogonal projection and minimum residual norm.

form' for the corresponding PDE. In the weak form, a derivative order balance is achieved. Integration by parts is equivalent to Green's identity in one dimension and if applied to Equation (1.42), one obtains the weak form:

$$\sum_j \left\{ q_j(t) \int_0^l \frac{dU_i}{dx} \left(EA \frac{dY_j}{dx} \right) dx + \ddot{q}_j(t) \int_0^l U_i m Y_j dx \right\} = P(t) \int_0^l U_i F_A(x) dx, i = 1,2,\ldots$$

$$\Rightarrow \sum_j \left\{ \mathcal{K}(U_i, Y_j) q_j + \mathcal{M}(U_i, Y_j) \ddot{q}_j \right\} = P\ell(U_i), i = 1,2,\ldots \tag{1.43}$$

Details on the derivation of the last equation are provided in Appendix 1 (under the item – Green's identity). In deriving the above equation, we assume that the forcing function $f_A(x,t)$ may be written in a variable separable form as $P(t)F_A(x)$. $\mathcal{K}(U_i, Y_j)$ and $\mathcal{M}(U_i, Y_j)$ are bilinear forms on $\mathcal{H} \times \mathcal{H}$ and $\ell(U_i)$, a linear form on $\mathcal{H}$ (see Appendix 1 for definitions of bilinear and linear forms). The integrals $\int_0^l \frac{dU_i}{dx} \left(EA \frac{dY_j}{dx} \right) dx$, $\int_0^l U_i m Y_j dx$ and $\int_0^l U_i F_A(x) dx$ in Equation (1.43) (once evaluated) yield the matrices $\mathcal{K}$, $\mathcal{M}$ and the vector ℓ respectively. This results in a set of ODEs the solution of which yields the generalized coordinates $q_j(t)$ and finally the required $\tilde{y}(\boldsymbol{x}, t)$ from Equation (1.41).

The well-posedness, uniqueness and stability of the solution to the weak form in Equation (1.43) are discussed in detail in Roy and Rao (2012). Here we bring in the notation $D^\alpha u$ for the α^{th} weak derivative of a function $u \in \mathcal{H}$ where $\alpha = 0,1,2,\ldots$. See Appendix 1 for the definition of weak derivatives. Note however that, for a sufficiently smooth u having derivatives of orders 1 to α in the classical sense, no distinctions need be made between classical and weak derivatives. The well-posedness requires that the basis and the test functions in the weak form belong to the Sobolev space $H^m(\mathfrak{V}) = \left\{ v : D^\alpha u \in L^2(\mathfrak{V}) \forall\, \alpha \text{such that} |\alpha| \le m \right\}$ (see Hilbert spaces in Appendix 1 and also Appendix 2 for details on Sobolev space). For the weak form in Equation (1.43), a choice of continuous and piece-wise linear functions $\Phi_j(x)$ and $\psi_j(x)$ (which belong to $H^1(0,1)$, even though they do not have classically defined first order derivatives) satisfies the smoothness requirement.

FEM

FEM may be viewed as either a Rayleigh-Ritz method with piecewise smooth trial functions or a Galerkin method with similar trial and test functions. Here, we start with a given discretization of the domain $\mathfrak{V}$ of interest into a set of 'non-overlapping' elements, $\left\{ \mathfrak{V}_i \right\}, i = 1,2,\ldots,N_e$ such that $\bigcup_{i=1}^{N_e} \mathfrak{V}_i = \mathfrak{V}$. By 'non-overlapping' elements, it means that two distinct elements $\mathfrak{V}_i$ and $\mathfrak{V}_j$ can at best have a common boundary for $i \neq j$. N_e stands for the number of elements in the finite element (FE) model. In the interior of each $\mathfrak{V}_i$, the trial/test functions are smooth. The lack of smoothness occurring at inter-element boundaries must be such that the overall approximation remains continuous (single valued) everywhere in $\mathfrak{V}$. The discretization described thus far is in fact referred to as semi-discretization (i.e., discretization of the spatial domain alone and not the time axis).

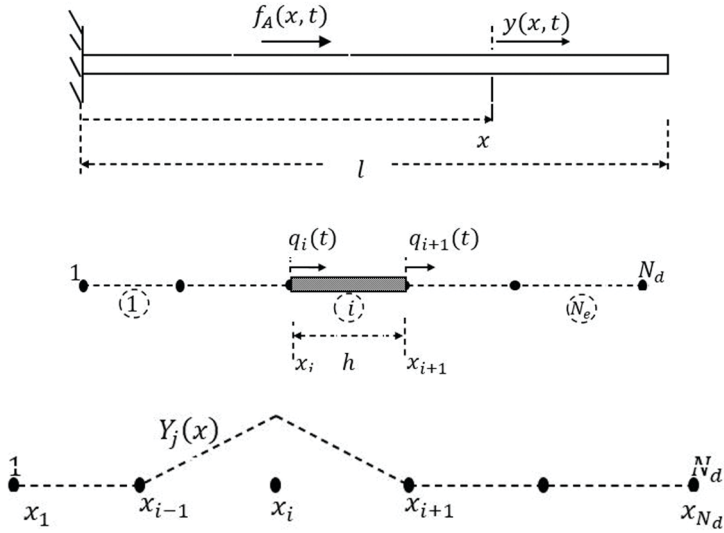


FIGURE 1.21 (a) Axially vibrating rod, (b) FEM semi-discretization – i^{th} element and nodal displacement functions $q_i(t)$ and $q_{i+1}(t)$ and (c) trial function $Y_j(x_i) = \delta_{ij}, i, j = 1,2,\ldots,N_d$.

Figure 1.21 shows an FE semi-discretization (also called FE mesh) of the vibrating rod of Example 1.1. For this specific system, $\mathfrak{V} = [0,l]$ and $\mathfrak{V}_i = [x_i, x_{i+1}]$ (see Figure 1.21b). The FE mesh consists of line elements with each element containing $n_e = 2$ nodes. Let N_d denote the total number of nodes in the discretization of $\mathfrak{V}$.

With the semi-discretization as in Figure 1.21, the axial displacement field component $y(x,t)$ at a generic location x is approximated as (similar to Equation 1.41):

$$\tilde{y}(x,t) = \sum_{j=1}^{N_d} Y_j(x) q_j(t) \tag{1.44}$$

In FEM, the trial functions $\{Y_j(x)\}$ form a polynomial basis set.[6] A polynomial basis set is called interpolating if for any node $i \in [1, N_d]$ with coordinate x_i, there exists only one basis function $Y_i(x)$ such that $Y_i(x_i) = 1$. Further, $Y_k(x_i) = 0$ for any node $k \neq i$ (Figure 1.21c). The interpolating functions $Y_j(x)$ are also referred to as shape functions. From Equation (1.44), we see that at the i^{th} node, $\tilde{y}(x_i,t) = \sum_{j=1}^{N_d} Y_j(x_i) q_j(t) = q_i(t)$ and similarly $\tilde{y}(x_{i+1},t) = \sum_{j=1}^{N_d} Y_j(x_{i+1}) q_j(t) = q_{i+1}(t)$ (Figure 1.21b). Thus, in the FE based semi-discretization, the generalized coordinates $q_j(t), j = 1,2,\ldots,N_d$ are immediately identifiable with the physically meaningful nodal displacements for the vibrating rod. This is unlike the standard Rayleigh-Ritz and weighted residuals methods discussed earlier. With the choice of test functions $U_j = Y_j$, the weak form (Equation 1.43) reduces to a system of linear coupled ODEs which can be written in a matrix form:

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{\mathcal{P}}(t),\ \boldsymbol{\mathcal{P}} \in \mathbb{R}^{N_d} \tag{1.45}$$

Here $\boldsymbol{M}$ and $\boldsymbol{K}$ are the (symmetric) mass and stiffness matrices in $\mathbb{R}^{N_d \times N_d}$ that result from the bilinear forms $\mathcal{M}\left(Y_k, Y_j\right) = \int_0^l m Y_k Y_j dx$ and $\mathcal{K}\left(Y_k, Y_j\right) = \int_0^l EAY_k' Y_j' \, dx$ respectively for $j,k = 1,2,\ldots,N_d$. Moreover, $\mathcal{P}_j = \int_0^l Y_j F_A dx$. The coupled ODEs in Equation (1.45) need to be solved for the unknown vector $\boldsymbol{q}(t) = \left(q_1, q_2, \ldots, q_{N_d}\right)^T$ to finally obtain $\tilde{y}(x,t)$ using Equation (1.44).

An added computational advantage with the FEM is the element-wise restriction of the shape functions (henceforth referred to as element shape functions) which in turn enables an element-wise splitting of the weak form. Specifically, the domain integration $\int_{\mathfrak{V}} d\mathfrak{V}$ in the weak form can be replaced by $\sum_i \int_{\mathfrak{V}_i} d\mathfrak{V}_i$.

Whilst the shape functions $Y_j(x)$ in Equation (1.44) directly yield the system matrices $\boldsymbol{M}$ and $\boldsymbol{K}$, element-based operations enable the computational scheme to be strictly modular and hence more easily implementable for complex systems (Bathe 1996, Hughes 2012).

1.6 CONSTRAINED OPTIMIZATION PROBLEMS AND OPTIMALITY CONDITIONS

All optimization problems are in general constrained (as formulated in Equations 1.3). From Equation (1.3b), it is evident that constraints of equality or inequality types may relate to complex relationships among the design variables. Further, these constraints may be linear or nonlinear. The optimality criteria for a constrained optimization problem are given by Karush-Kuhn-Tucker (KKT) conditions (Karush 1939, Khun and Tucker 1951) which are necessarily satisfied by a local optimum. These optimality conditions are explained in the following sub-sections via intuitive arguments supported by a graphical description wherever necessary. For an unconstrained optimization problem, vanishing of the gradient $\nabla f(\boldsymbol{x})$ at an optimum point is necessary. But it may not be the case when constraints are present. Note that the gradient $\nabla f(\boldsymbol{x})$ points towards the direction of steepest ascent for the function $f(\boldsymbol{x})$ while $-\nabla f(\boldsymbol{x})$ points towards the direction of steepest descent. This is clarified by a Taylor expansion of $f(\boldsymbol{x} + \alpha \boldsymbol{d})$ around $\boldsymbol{x}$ with $\alpha \in \mathbb{R}^+$ denoting a step size in the direction $\boldsymbol{d}$.

$$f(\boldsymbol{x} + \alpha \boldsymbol{d}) = f(\boldsymbol{x}) + \alpha \left\{\nabla f(\boldsymbol{x})\right\}^T \boldsymbol{d} + \mathrm{O}(\alpha^2) \tag{1.46}$$

$\mathrm{O}(\alpha^2)$ in the last equation stands for the order of approximation* and signifies that the remainder terms tend to zero faster than as a linear function of α as $\alpha \to 0$. Thus, for small α, one has:

---

* Orders of approximation
Big 'O' and small 'o' notations are used to describe the asymptotic behaviour of functions.
Big 'O' *notation:* $F(x) = \mathrm{O}(\boldsymbol{G}(\boldsymbol{x}))$ means that as $x \to \infty$, there exist constants N and K such that $F(x) \leq K G(x)$ for all $x > N$. That is, $F(x)$ grows no faster than $G(x)$.

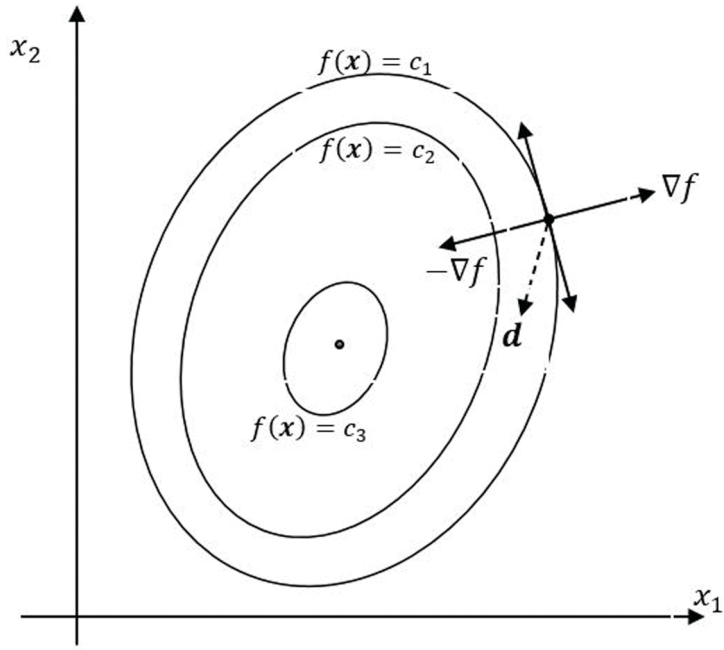


FIGURE 1.22 Geometric significance of the gradient vector and directions of steepest ascent and descent.

$$\Delta f = f\left(\boldsymbol{x} + \alpha \boldsymbol{d}\right) - f\left(\boldsymbol{x}\right) \approx \alpha \left\{\nabla f\left(\boldsymbol{x}\right)\right\}^T \boldsymbol{d} \tag{1.47}$$

If the directions of $\boldsymbol{d}$ and $-\nabla f\left(\boldsymbol{x}\right)$ coincide, then $\Delta f < 0$ which shows that $-\nabla f\left(\boldsymbol{x}\right)$ is indeed the direction of steepest descent. This is the basis for optimization by steepest descent (details provided in Chapter 2). While a normal to the gradient vector $\nabla f\left(\boldsymbol{x}\right)$ or $-\nabla f\left(\boldsymbol{x}\right)$ lies in the plane tangent to the equi-potential (equi-cost) curves (see Figure 1.22 for a two-dimensional problem), a movement along a direction $\boldsymbol{d}$ lying to the side of this plane containing $-\nabla f\left(\boldsymbol{x}\right)$ decreases $f\left(\boldsymbol{x}\right)$. The vector $\boldsymbol{d}$ is thus a descent direction if $\nabla f^T \boldsymbol{d} < 0$ and an ascent direction otherwise. Convergence is extremely slow for the steepest descent method while handling functions with valleys and troughs. The derivative methods of conjugate gradient (Fletcher and Reeves 1964) and quasi-Newton (Nocedal and Wright 2006) which are also described in Chapter 2, use descent directions $\boldsymbol{d}$ suitably modified to achieve better performance (than the method of steepest descent).

Small 'o' *notation:* $F(x) \in \mathrm{o}\left(G(x)\right)$ means that as $x \to \infty$, there exist constants N and K such that for all $x > N$, one has $\left|F(x)\right| < K\left|G(x)\right|$. That is, $F(x)$ grows much slower than $G(x)$. If $G(x) \neq 0$ ' this is equivalent to $\lim_{x \to \infty} F(x) / G(x) = 0$

1.6.1 Optimization Problem with Equality Constraints

We first consider an optimization problem with equality constraints. Specifically Figure 1.23 shows a situation with a single equality constraint in a 2-dimensional optimization setting. The problem is to:

$$\text{Minimize } f(\boldsymbol{x})$$

$$\text{s.t } h(\boldsymbol{x}) = 0 \tag{1.48}$$

The gradient vector $\nabla h(\boldsymbol{x})$ in the figure has the same interpretation as that of $\nabla f(\boldsymbol{x})$. It is clear that a constrained optimum – perhaps a local one – must lie on the constraint curve. Suppose that one arrives at $\boldsymbol{x} = (x_1, x_2)^T$ - marked '1' in the figure – during the optimization process with respective gradient vectors as shown. The gradient vector $-\nabla f$ still has a component pointing towards a direction perpendicular to Δh implying that there is scope to improve the minimum by traversing along the constraint curve towards the right.

A similar situation arises when the current point reaches the one marked 3 in the figure. In this case, further exploration for a better result may be required by moving up the constraint curve towards the left. However, when the point marked 2 is reached, directions of the gradient vectors coincide excluding any scope for further

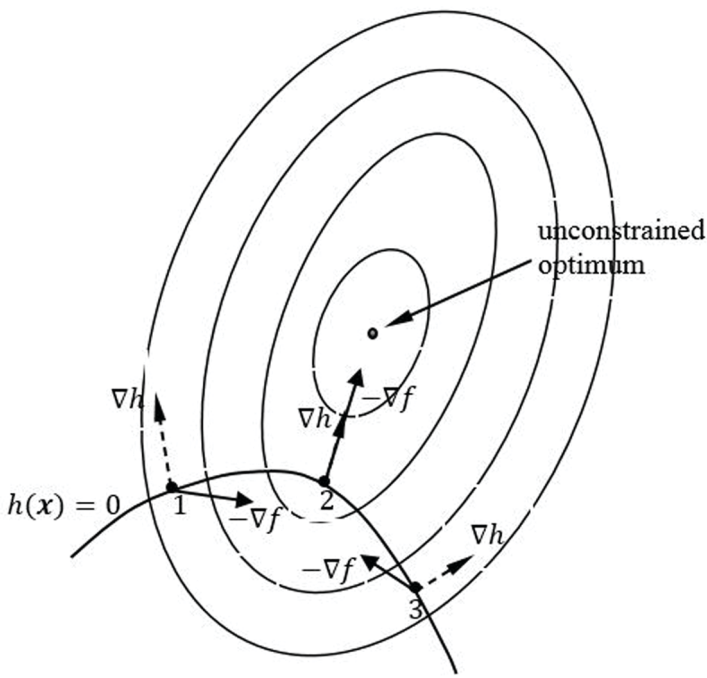


FIGURE 1.23 A constrained optimization problem with an equality constraint.

improvement. That is, the point is the local optimum $\boldsymbol{x}^*$ and the gradient vector ∇h is parallel to ∇f at this point with the result $-\nabla f(\boldsymbol{x}^*) = \mu \nabla h(\boldsymbol{x}^*)$ with μ being a scalar constant of proportionality. In this case of an equality constraint, whether ∇h is parallel or anti-parallel with $-\nabla f(\boldsymbol{x}^*)$ is inconsequential; the reason being that $\boldsymbol{x}^*$ always lies on $h(\boldsymbol{x}) = 0$. It implies that μ may be > 0 or < 0. Note that $h(\boldsymbol{x}^*) = 0$.

Consider the case of two equality constraints as shown in Figure 1.24 for the 2-dimensional optimization problem. Following similar arguments as in the case of a single constraint, one finds that the points marked 4 and 5 on the second equality constraint $h_2(\boldsymbol{x}) = 0$ need correction so as to be brought closer to the local optimum, just as the points marked 1 and 3 on the first constraint $h_1(\boldsymbol{x}) = 0$.

At the point marked 2 common to the two constraints (where both equality constraints are satisfied), no further improvement is possible leading to the conclusion that $\boldsymbol{x}^*$ is reached and $-\nabla f$ is a linear combination of the gradient vectors Δh_1 and ∇h_2, i.e. $-\nabla f(\boldsymbol{x}^*) = \mu_1 \nabla h_1(\boldsymbol{x}^*) + \mu_2 \nabla h_2(\boldsymbol{x}^*)$ with $\mu_1, \mu_2 \in \mathbb{R}$. In arriving at the result, linear independence of the gradient vectors ∇h_1 and ∇h_2 must be assumed. The requirement of linear independence essentially ensures that the constraints are unique and none of them is redundant. If we generalize the result for an n-dimensional

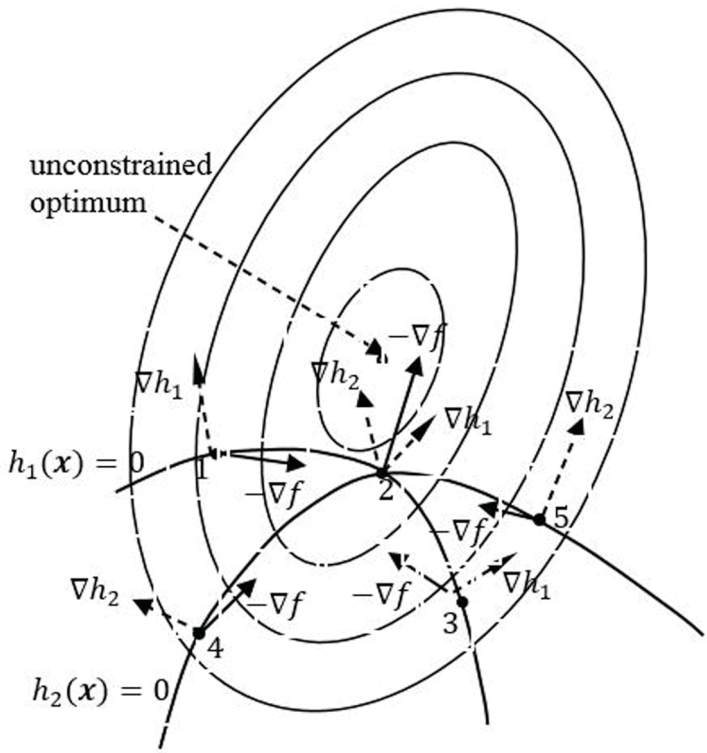


FIGURE 1.24 A constrained optimization problem with two equality constraints.

optimization problem with $l \geq 2$ linearly independent equality constraints, the local optimizer satisfies the conditions:

$$-\nabla f\left(\boldsymbol{x}^*\right) = \sum_{i=1}^{l} \mu_i \nabla h_i\left(\boldsymbol{x}^*\right)$$

$$\Rightarrow \nabla f\left(\boldsymbol{x}^*\right) + \sum_{i=1}^{l} \mu_i \nabla h_i\left(\boldsymbol{x}^*\right) = 0, \mu_i \in \mathbb{R} \tag{1.49a}$$

and

$$h_i\left(\boldsymbol{x}^*\right) = 0,\ i = 1,2,\ldots,l \tag{1.49b}$$

The above are the KKT conditions for the optimization problem with equality constraints. These conditions are used to solve a set of $n+l$ equations for the $n+l$ unknowns $\boldsymbol{x}^* \in \mathbb{R}^n$ and $\boldsymbol{\mu} \in \mathbb{R}^l$.

Equations (1.49) indicate that one can formulate the optimization problem with equality constraints as an unconstrained one by introducing a Lagrangian† defined by:

$$L\left(\boldsymbol{x}, \boldsymbol{\mu}\right) = f\left(\boldsymbol{x}\right) + \sum_{i=1}^{l} \mu_i h_i\left(\boldsymbol{x}\right) \tag{1.50}$$

---

† Lagrangian

Here the use of the word Lagrangian may be attributed to the association it has with the familiar notion of functional optimization discussed in earlier sections of this Chapter. For instance, in the context of variational calculus applied to the brachistochrone problem and Hamiltonian mechanics in Sections 1.4 and 1.5, the respective action integrals in Equations (1.13) and (1.34) involving the Lagrangian led to the EL equations via a stationarity principle. Similarly, based on Equation (1.50), one may form an action integral:

$$I = \int_{t_1}^{t_2} L\left(\boldsymbol{x}, \boldsymbol{\mu}\right) dt \tag{i}$$

Here t_1 and t_2 represent pseudo time instants parametrizing the iterative steps involved in an optimization process. Using the stationarity of the functional, we arrive at (1.49) by a straightforward exercise:

$$\delta I = 0 \Rightarrow \int_{t_1}^{t_2} \left\{ \left(\nabla f_{\boldsymbol{x}} + \sum_{i=1}^{l} \mu_i \nabla h_{i,\boldsymbol{x}} \right) \delta \boldsymbol{x} + \sum_{i=1}^{l} h_i \delta \mu_i \right\} dt = \boldsymbol{0} \tag{ii}$$

$$\Rightarrow \nabla f_{\boldsymbol{x}} + \sum_{i=1}^{l} \mu_i \nabla h_{i,\boldsymbol{x}} = 0 \text{ and } h_i = 0, i = 1,2,..,l \text{ with } \mu_i \in \mathbb{R} \setminus 0$$

The above result is obtained by the localization theorem (Rudin 1976). According to this theorem, if Φ is a continuous field (scalar or vector) on V and if, for all closed sets $B \subset V$, $\int_B \Phi d\zeta = 0$, then $\Phi = 0$ for all $u \in V$.

One readily finds the necessary optimality conditions in the form of vanishing gradients as:

$$\nabla_x L = 0 \tag{1.51a}$$

and

$$\nabla_\mu L = 0 \tag{1.51b}$$

∇_x and ∇_μ denote directional derivatives along the vectors $\boldsymbol{x}$ and $\boldsymbol{\mu}$ respectively. These conditions are identical to those in Equation (1.49). $\mu_i, i = 1,2,\ldots,l$ are called Lagrange multipliers. This is known as the method of Lagrange multipliers.

1.6.2 Optimization Problem with Inequality Constraints

Now, consider the optimization problem with an inequality constraint (Figure 1.25):

$$\text{minimize } f(\boldsymbol{x})$$

$$\text{s.t. } g(\boldsymbol{x}) \le 0 \tag{1.52}$$

As shown in Figure 1.25, the feasible region is below the hatched curve $g(\boldsymbol{x}) = 0$. In case the inequality constraint is satisfied by the unconstrained optimum as depicted in Figure 1.25a, the constraint is said to be no longer 'binding', or it is in fact a slack constraint. It follows that the local optimizer is the same as that of an unconstrained optimization problem. In such a case, the necessary condition for the local optimizer is simply $-\nabla f(\boldsymbol{x}^*) = 0$ and this can be construed as the KKT condition $\nabla f(\boldsymbol{x}^*) = \lambda g(\boldsymbol{x}^*)$ with $\lambda = 0$.

In Figure 1.25b, the inequality constraint is binding and active so that the local optimum must lie on the constraint boundary. Putting forward the same arguments as in the case of equality constraints, one locates the local optimizer $\boldsymbol{x}^*$ at the point marked $'2'$ where the gradient vectors $-\nabla f$ and ∇g are parallel to each other. However, unlike the case of equality constraints, a point $\boldsymbol{x}$ is not a local optimizer if $-\nabla f$ and ∇g are anti-parallel. The reason is that with $g(\boldsymbol{x})$ being an inequality constraint, $-\nabla f$ at the constrained minimum $\boldsymbol{x}^*$ will always point towards the unconstrained minimum. Thus, to locate $\boldsymbol{x}^*$ at point 2, it is required to have the KKT condition as $-\nabla f(\boldsymbol{x}^*) = \lambda \nabla g(\boldsymbol{x}^*)$ with $\lambda > 0$. Combining the two possible cases of the active and slack inequality constraints, the necessary optimality criteria are given by the KKT condition $-\nabla f(\boldsymbol{x}^*) = \lambda \nabla g(\boldsymbol{x}^*)$, $\lambda \ge 0$. The inequality constraint is satisfied at the optimum, i.e., $g(\boldsymbol{x}^*) \le 0$. As with equality constraints, we assume that, for multiple inequality constraints, the gradient vectors of the active constraints are linearly independent. With a set of m (> 1) inequality constraints, we can state the KKT conditions as:

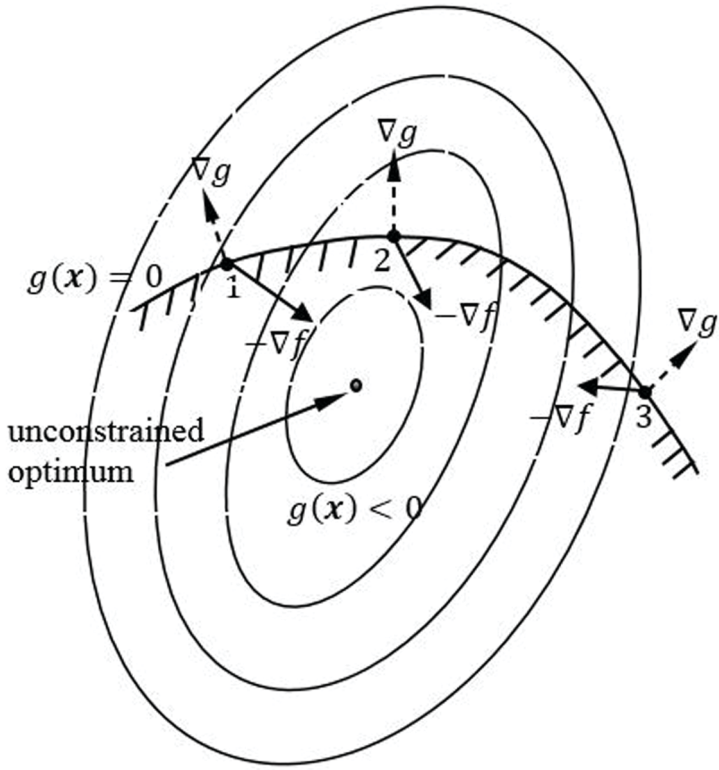


FIGURE 1.25a Constrained optimization problem with an inequality constraint; (a) case of a slack inequality constraint (not binding), hence solution search in the interior of $g(\boldsymbol{x}) < 0$; (b) case of an active inequality constraint and solution search on the surface of $g(\boldsymbol{x}) = 0$, $\boldsymbol{x}^*$ denotes the local optimum.

$$-\nabla f\left(\boldsymbol{x}^*\right) = \sum_{i=1}^{m} \lambda_i \nabla g_i\left(\boldsymbol{x}^*\right)$$

$$\Rightarrow \nabla f\left(\boldsymbol{x}^*\right) + \sum_{i=1}^{m} \lambda_i \nabla g_i\left(\boldsymbol{x}^*\right) = 0, \tag{1.53a}$$

$$\lambda_i \geq 0, i = 1, 2, \ldots, m \tag{1.53b}$$

and

$$g_i\left(\boldsymbol{x}^*\right) \leq 0, i = 1, 2, \ldots, m \tag{1.53c}$$

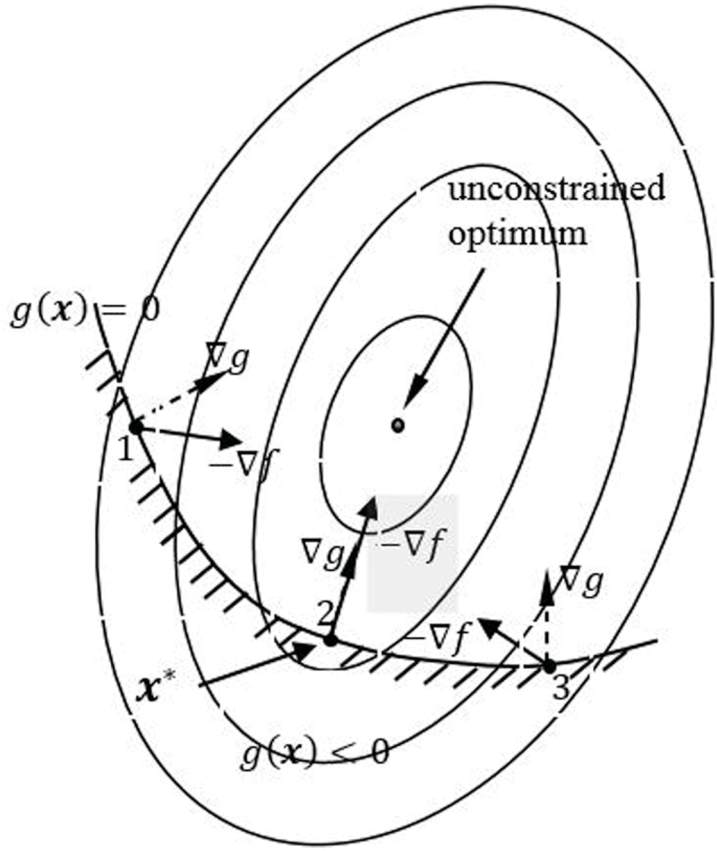


FIGURE 1.25b (Continued)

Since only active constraints will have non-zero multipliers λ_i, a complementary slackness condition is added to the above as:

$$\lambda_i g_i\left(\boldsymbol{x}^*\right) = 0, i = 1,2,\ldots,m \tag{1.53d}$$

One can define a Lagrangian by recasting the above problem with constraints as an unconstrained optimization problem. By the method of Lagrange multipliers, the optimality conditions – same as the KKT conditions in Equation (1.53), may be derived. This is described in the next sub-section where a general optimization problem with both equality and inequality constraints is discussed.

Example 1.2. We consider the constrained optimization problem illustrated in Figure 1.1 and examine the relevant KKT conditions.

Solution. This is a two-dimensional maximization problem with the design variable vector $\boldsymbol{x} = \left(x_1, x_2\right)^T$ and a linear inequality constraint. The problem is stated as:

$$\text{maximize } f\left(\boldsymbol{x}\right) = \sqrt{x_1 x_2} \tag{1.54a}$$

$$\text{s.t } g\left(\boldsymbol{x}\right) = 3x_1 + x_2 - 400 \leq 0 \tag{1.54b}$$

Being a maximization problem, it requires that $\nabla f\left(\boldsymbol{x}^*\right)$ and $\nabla g\left(\boldsymbol{x}^*\right)$ must be parallel at the optimal point (see the discussion on the minimization problem corresponding to Figure 1.25b). Thus $\nabla f\left(\boldsymbol{x}^*\right) = \lambda \nabla g\left(\boldsymbol{x}^*\right)$ with $\lambda > 0$.
The Lagrangian $L\left(\boldsymbol{x}, \lambda\right)$ is:

$$L\left(\boldsymbol{x}, \lambda\right) = \sqrt{x_1 x_2} - \lambda\left(3x_1 + x_2 - 400\right) \tag{1.55}$$

With $m = 1$, the KKT conditions are thus given by:

$$\nabla f\left(\boldsymbol{x}^*\right) - \lambda \nabla g\left(\boldsymbol{x}^*\right) = 0 \tag{1.56a}$$

$$\lambda \geq 0 \tag{1.56b}$$

$$g\left(\boldsymbol{x}^*\right) \leq 0 \tag{1.56c}$$

and

$$\lambda g\left(\boldsymbol{x}^*\right) = 0 \tag{1.56d}$$

These are the necessary conditions to be satisfied at the optimal point. The optimal point has already been found for this problem graphically (see Figure 1.1) as $\boldsymbol{x}^* = (x_1 = 66.5, x_2 = 200)^T$. The gradient vectors ∇f and ∇g at $\boldsymbol{x}^*$ are given by:

$$\nabla f\left(\boldsymbol{x}^*\right) = \begin{pmatrix} 0.867 \\ 0.288 \end{pmatrix} \text{and} \nabla g = \left(\boldsymbol{x}^*\right) \begin{pmatrix} 3 \\ 1 \end{pmatrix} \tag{1.57}$$

Substitution of these vectors in Equation (1.56a) gives $\lambda \approx 0.288$ which is positive showing that the constraint is active. Further, $\boldsymbol{x}^*$ satisfies the constraint (1.56c) with strict equality and hence also Equation (1.56d) with $\lambda \neq 0$.

■

Example 1.3. Figure 1.2 is another illustration of a constrained optimization problem with an inequality constraint. We verify the necessary KKT conditions for optimality.

Solution: The optimization problem if written in the standard form of Equation (1.52) is given by:

$$\text{minimize } f(\boldsymbol{x}) = 4x_1 + x_2 \tag{1.58a}$$

$$\text{s.t } g(\boldsymbol{x}) = -\sqrt{x_1 x_2} + 100 \leq 0 \tag{1.58b}$$

From Figure 1.2, the optimal solution $\boldsymbol{x}^* = (x_1 = 50, x_2 = 200)^T$. The gradient vectors at $\boldsymbol{x}^*$ are given by:

$$-\nabla f(\boldsymbol{x}^*) = \begin{pmatrix} -4 \\ -1 \end{pmatrix} \text{and } \nabla g(\boldsymbol{x}^*) = \begin{pmatrix} -1 \\ -0.25 \end{pmatrix} \tag{1.59}$$

$\lambda = 4$ satisfies the KKT condition in Equation (1.56a) with $m = 1$. While the strict equality condition (1.56c) is also satisfied, the non-zero positive λ satisfies the complementary slack condition (1.56d).

■

1.6.3 Optimization Problem with Both Equality and Inequality Constraints

A general optimization problem with both equality and inequality constraints is of the form:

$$\text{minimize } f(\boldsymbol{x})$$

$$\text{s.t } h_i(\boldsymbol{x}) = 0, i = 1, 2, \ldots, l$$

$$\text{and } g_j(\boldsymbol{x}) \leq 0, j = 1, 2, \ldots, m \tag{1.60}$$

Combining the arguments in the earlier subsections for the cases of equality and inequality constraints, one arrives at the optimality (KKT) conditions for the present optimization problem as:

$$-\nabla f(\boldsymbol{x}^*) = \sum_{i=1}^{l} \mu_i \nabla h_i(\boldsymbol{x}^*) + \sum_{j=1}^{m} \lambda_j \nabla g_j(\boldsymbol{x}^*) \tag{1.61a}$$

$$h_i(\boldsymbol{x}^*) = 0,\ i = 1, 2, \ldots, l, \tag{1.61b}$$

$$g_j(\boldsymbol{x}^*) \leq 0, j = 1, 2, \ldots, m \text{ and} \tag{1.61c}$$

$$\mu_i \in \mathbb{R}, i = 1, 2, \ldots, l,\ \lambda_j \geq 0, j = 1, 2, \ldots, m \tag{1.61d}$$

$$\lambda_j g_j(\boldsymbol{x}) = 0, j = 1, 2, \ldots, m \tag{1.61e}$$

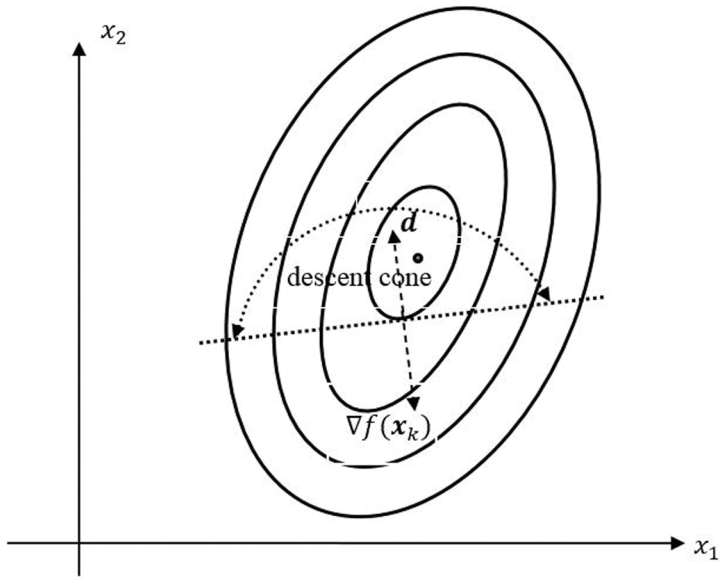


FIGURE 1.26a Descent cone and descent direction $\boldsymbol{d} = -\nabla f\left(x_k\right)$.

The last one (1.61e) is the complementary slackness condition to ensure that only active constraints will have non-zero multipliers λ_j. The above KKT conditions, arrived at purely by geometric considerations (Figures 1.24 and 1.25), may be interpreted algebraically using Farkas' lemma (Appendix 1). To this end, we define the feasible space $\mathcal{D} = \left\{\boldsymbol{x} : g_j\left(\boldsymbol{x}\right) \le 0,\ j = 1,2,\ldots,m,\ h_i\left(\boldsymbol{x}\right) = 0,\ i = 1,2,\ldots,l\right\} \subset \Xi$ where Ξ is the design space. With $\boldsymbol{x} \in \mathcal{D}$, let $F = \left\{\boldsymbol{d} : \nabla f^T \boldsymbol{d} < 0\right\}$ be the cone of descent directions (Figure 1.26a) of f at $\boldsymbol{x}$, $H = \left\{\boldsymbol{d} : \nabla h_i^T \boldsymbol{d} = 0\ \forall\, i = 1,2,\ldots,l\right\}$ a set of tangent directions of equality constraints and $G = \left\{\boldsymbol{d} : \nabla g_j^T \boldsymbol{d} < 0\ \forall\ j = 1,2,\ldots,m\right\}$ the cone of feasibility directions for the inequality constraints. Here a cone $C \subset \mathbb{R}^n$ is a set with the property that for every $\boldsymbol{x} \in C$, $\alpha \boldsymbol{x} \in C$ for any $\alpha > 0$. Figure 1.26b shows a feasibility cone G for the case of two active inequality constraints. If $\boldsymbol{x}^*$ is the local optimizer, the geometric (necessary) optimality condition in Equation (1.61a) is equivalent to $F \cap H \cap G = \phi$ in that there exists no vector $\boldsymbol{d}$ at $\boldsymbol{x}^*$ which is both a descent and a feasible direction. According to Farkas' lemma also, we have the following assertion:

Exactly one of the following systems has a solution:

1) $\nabla f\left(\boldsymbol{x}^*\right) + \sum_{i=1}^{l} \mu_i \nabla h_i\left(\boldsymbol{x}^*\right) + \sum_{j=1}^{m} \lambda_j \nabla g_j\left(\boldsymbol{x}^*\right) = 0, \lambda_j \ge 0, j = 1,2,\ldots,m, \mu_i \in \mathbb{R}, i = 1,2,\ldots,l$

2) $\nabla f^T\left(\boldsymbol{x}\right)\boldsymbol{d} < 0,\ \nabla h^T\left(\boldsymbol{x}\right)\boldsymbol{d} = 0,\ \nabla g^T\left(\boldsymbol{x}\right)\boldsymbol{d} \le 0$ (1.62a,b)

The above assertion implies that the set $S = \left\{\boldsymbol{d} : \nabla \boldsymbol{f}^{\boldsymbol{T}}\left(\boldsymbol{x}\right)\boldsymbol{d} < 0, \nabla \boldsymbol{h}^{\boldsymbol{T}}\left(\boldsymbol{x}\right)\boldsymbol{d} = 0, \nabla \boldsymbol{g}^{\boldsymbol{T}}\left(\boldsymbol{x}\right)\boldsymbol{d} \le 0\right\}$ is empty if and only if $\nabla f\left(\boldsymbol{x}^*\right) + \sum_{i=1}^{l} \mu_i \nabla h_i\left(\boldsymbol{x}^*\right) + \sum_{j=1}^{m} \lambda_j \nabla g_j\left(\boldsymbol{x}^*\right) = 0$ for some $\lambda_j \ge 0$ and $\mu_i \in \mathbb{R}$ which

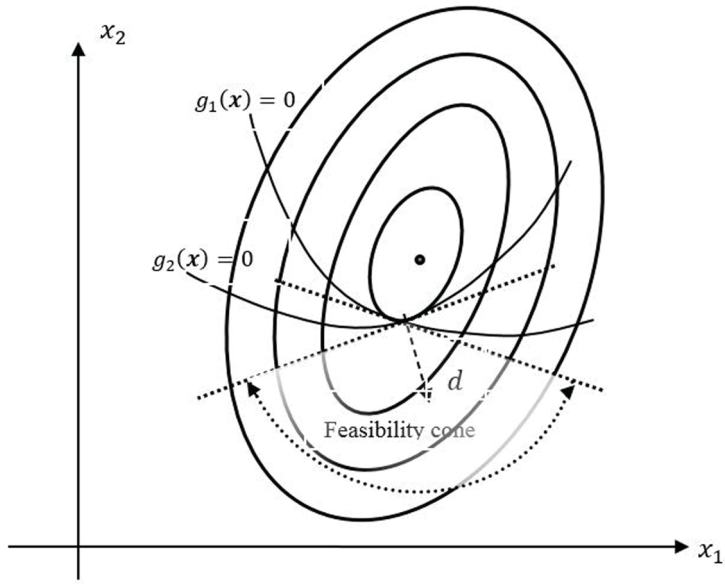


FIGURE 1.26b Feasibility cone and feasibile direction $\nabla f\left(x_k\right)$.

is the necessary KKT condition for the general optimization problem in Equation (1.60). Equation (1.61a) suggests that a general minimization problem with both equality and inequality constraints can be stated in terms of a Lagrangian as:

$$L\left(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu}\right) = f\left(\boldsymbol{x}\right) + \sum_{i=1}^{l} \mu_i h_i\left(\boldsymbol{x}\right) + \sum_{i=1}^{m} \lambda_i g_i\left(\boldsymbol{x}\right) \tag{1.63}$$

The Lagrangian is a function of $\boldsymbol{x} = \left(x_1, x_2, \ldots, x_n\right)^T$, $\boldsymbol{\lambda} = \left(\lambda_1, \lambda_2, \ldots, \lambda_m\right)^T$ and $\boldsymbol{\mu} = \left(\mu_1, \mu_2, \ldots, \mu_l\right)^T$. As an unconstrained optimization problem, the above form has the following KKT condition necessary for optimality:

$$\nabla L = \begin{pmatrix} \nabla_x L \\ \nabla_\lambda L \\ \nabla_\mu L \end{pmatrix} = \mathbf{0} \tag{1.64}$$

which are the same as those in Equation (1.61). For a maximization problem, the Lagrangian in Equation (1.63) may be taken as:

$$L\left(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu}\right) = f\left(\boldsymbol{x}\right) - \sum_{i=1}^{l} \mu_i h_i\left(\boldsymbol{x}\right) - \sum_{j=1}^{m} \lambda_j g_j\left(\boldsymbol{x}\right) \tag{1.65}$$

The change in the sign of the last two terms on the RHS of Equation (1.65) follows the fact that at the optimal point the (positive) gradient $\nabla f(\boldsymbol{x}^*)$ must be a linear combination of the gradient vectors $\nabla h_i, i = 1,2,\ldots,l$ and $\nabla g_j, j = 1,2,\ldots,m$.

The KKT conditions are equally applicable to an unconstrained optimization problem. This is equivalent to the case where no constraint is active. Thus, all the Lagrange multipliers are zero leading to the only KKT condition $\nabla L = \nabla_x f = 0$.

1.6.4 Sufficient Conditions of Optimality for a Constrained Optimization Problem

The KKT conditions stated in the previous sub-sections are necessary for obtaining a local optimum. If, for example, a point $\boldsymbol{x}^*$ satisfies the KKT conditions for a minimum, it can be a local minimum or a saddle point (Appendix 1). The case is similar with a maximum. As in unconstrained optimization, the sufficiency condition for a local optimum in the presence of constraints depends on the second order derivatives. Referring to the optimization problem in Equation (1.63), one has the Hessian matrix $\nabla^2 L(\boldsymbol{x}^*)$ given by:

$$\nabla^2 L(\boldsymbol{x}^*) = \nabla^2 f(\boldsymbol{x}) + \sum_{i=1}^{l} \mu_i \nabla^2 h_i(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(\boldsymbol{x}) \tag{1.66}$$

If $\nabla^2 L(\boldsymbol{x}^*)$ is positive definite on the tangent subspace $\mathcal{T}$ of the equality and active inequality constraints, the point $\boldsymbol{x}^*$ is a strict local minimum. The tangent space $\mathcal{T}$ is defined as:

$$\mathcal{T} = \left\{ \boldsymbol{y} : \nabla h_i(\boldsymbol{x}^*)^T \boldsymbol{y} = 0, \nabla g_j(\boldsymbol{x}^*)^T \boldsymbol{y} = 0 \text{ and } g_j(\boldsymbol{x}^*) = 0 \text{ with } \lambda_j > 0 \right\} \tag{1.67}$$

The Hessian is positive definite if $\boldsymbol{y}^T \nabla^2 L(\boldsymbol{x}^*) \boldsymbol{y} > 0 \; \forall \boldsymbol{y} \neq 0$. This second-order condition implies that the objective and the feasible domains are locally convex at the optimal point. A similar condition may be stated for a strict local maximum using the Lagrangian in Equation (1.65) in which case $\nabla^2 L(\boldsymbol{x}^*)$ needs to be negative definite on the tangent subspace $\mathcal{T}$, i.e. $\boldsymbol{y}^T \nabla^2 L(\boldsymbol{x}^*) \boldsymbol{y} < 0 \; \forall \boldsymbol{y} \neq 0$.

Example 1.4. We check the sufficiency condition for the optimization problems illustrated in Figures 1.1 and 1.2 (see corresponding Examples 1.2 and 1.3).

Solution. Referring to the Example 1.2 and Figure 1.1, we have the optimal point $\boldsymbol{x}^*$ for the maximization problem as $(66.5, 200)^T$. From the Lagrangian in Equation (1.55), one obtains:

$$\nabla^2 L(\boldsymbol{x}) = \begin{bmatrix} -\dfrac{1}{4x_1}\sqrt{\dfrac{x_2}{x_1}} & \dfrac{1}{4}\dfrac{1}{\sqrt{x_1 x_2}} \\ \dfrac{1}{4}\dfrac{1}{\sqrt{x_1 x_2}} & -\dfrac{1}{4x_2}\sqrt{\dfrac{x_1}{x_2}} \end{bmatrix}$$

$$\Rightarrow \nabla^2 L(\boldsymbol{x}^*) = \begin{bmatrix} -0.00652 & 0.00217 \\ 0.00217 & -0.000721 \end{bmatrix} \tag{1.68}$$

With $g(\boldsymbol{x}^*) = 0$ and $\nabla g(\boldsymbol{x}^*) = \begin{pmatrix} 3 & 1 \end{pmatrix}^T$, we find that $\boldsymbol{y} = \begin{pmatrix} -1 & 3 \end{pmatrix}^T$ is a vector in the tangent plane $\mathcal{T} = \left\{ \boldsymbol{y} : \nabla g(\boldsymbol{x}^*)^T \boldsymbol{y} = 0 \text{ and } g(\boldsymbol{x}^*) = 0 \text{ with } \lambda = 3 > 0 \right\}$ (see Figure 1.27a).

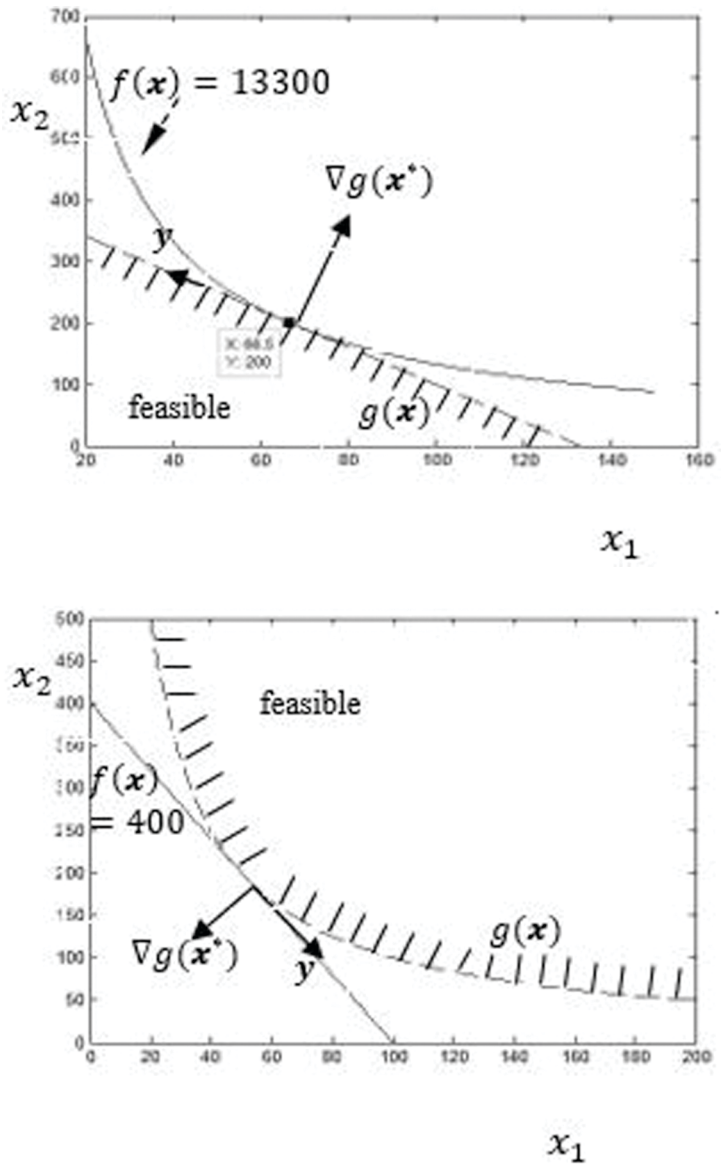


FIGURE 1.27 Tangent plane and second-order sufficient condition: (a) maximization problem in Example 1.2 and (b) minimization problem in Example 1.3.

Here with only one inequality constraint which is linear, the tangent plane coincides with the constraint graph itself. Note that $\boldsymbol{y}^T \nabla^2 L(\boldsymbol{x}^*)\boldsymbol{y} = -0.026 < 0$ and thus the Hessian is negative definite indicating that the optimal point is a strict maximum.

In Example 1.3 (corresponding to Figure 1.2), we have a minimization problem. The Lagrangian and the inequality constraint $g(\boldsymbol{x})$ are given by:

$$L(\boldsymbol{x}, \lambda) = 4x_1 + x_2 + \lambda\left(-\sqrt{x_1 x_2} + 100\right) \tag{1.69a}$$

and

$$g(\boldsymbol{x}) = -\sqrt{x_1 x_2} + 100 \tag{1.69b}$$

The optimal point $\boldsymbol{x}^*$ is $(50, 200)^T$ and the Lagrange multiplier $\lambda = 4$. We have:

$$\nabla^2 L(\boldsymbol{x}) = \begin{bmatrix} \dfrac{\lambda}{4x_1}\sqrt{\dfrac{x_2}{x_1}} & -\dfrac{1}{4}\dfrac{\lambda}{\sqrt{x_1 x_2}} \\ -\dfrac{1}{4}\dfrac{\lambda}{\sqrt{x_1 x_2}} & \dfrac{\lambda}{4x_2}\sqrt{\dfrac{x_1}{x_2}} \end{bmatrix}$$

$$\Rightarrow \nabla^2 L(\boldsymbol{x}^*) = \begin{bmatrix} 0.04 & -0.01 \\ -0.01 & 0.0025 \end{bmatrix} \tag{1.70}$$

It is also clear that $g(\boldsymbol{x}^*) = 0$ and $\nabla g(\boldsymbol{x}^*) = (-1 \quad -0.25)^T$. $\boldsymbol{y} = (1 \ -4)^T$ is a vector in the tangent plane $\mathcal{T} = \left\{\boldsymbol{y} : \nabla g(\boldsymbol{x}^*)^T \boldsymbol{y} = 0 \text{ and } g(\boldsymbol{x}^*) = 0 \text{ with } \lambda = 4 > 0\right\}$ (see Figure 1.27b). With only one inequality constraint, the tangent plane is the line of tangent to the nonlinear constraint. If we verify the sufficiency condition, we get $\boldsymbol{y}^T \nabla^2 L(X)\boldsymbol{y} = 0.16 > 0$; thus, the Hessian is positive definite and the optimal point $\boldsymbol{x}^*$ a strict minimum.

■

1.7 FUNCTIONAL OPTIMIZATION AND OPTIMAL CONTROL

Optimal control (Kirk 1970, Meirovitch 1990, Liberzon 2012) is a constrained optimization problem with the constraints being non-holonomic‡ and expressed in the

‡ non-holonomic constraints
Given the coordinates $x_i, i = 1, 2, \ldots, n$ describing a system, a constraint that is expressible in the form $f(x_1, x_2, \ldots, x_n, t) = 0$, where f is a smooth function not involving derivatives with respect to t, is known as a holonomic constraint. Constraints that are not expressible in the above form are non-holonomic.

form of DEs. It involves minimizing or maximizing a performance index. The index may aim at minimizing an error between the desired system output and the actual output of the system whose dynamics is described by $\frac{dx}{dt} = \dot{x} = f(x,u,t)$.

Here $x(t)$ is the state variable and $u(t)$ the control input variable guiding the system to reach the target. In general, for an *n*-dimensional system, an optimal control problem may be stated as:

$$\text{minimize } J = h\left(\boldsymbol{x}\left(t_f\right),t_f\right) + \int_{t_0}^{t_f} L\left(\boldsymbol{x}(t),\boldsymbol{u}(t),t\right)dt$$

$$\text{s.t } \dot{\boldsymbol{x}} = \boldsymbol{F}\left(\boldsymbol{x}(t),\boldsymbol{u}(t),t\right) \tag{1.71a,b}$$

where t_0 is the initial time and t_f the final time; $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{u} \in \mathbb{R}^m$. $\boldsymbol{F}(\boldsymbol{x},\boldsymbol{u},t) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^+ \to \mathbb{R}^n$ is an n dimensional vector function. $h(\boldsymbol{x},t): \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}$ and $L(\boldsymbol{x},\boldsymbol{u},t): \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^+ \to \mathbb{R}$ are scalar functions. The first term in the performance index J may be considered as a final cost and the second term as a running cost; the task is to arrive at an admissible control $\boldsymbol{u}^*$ and admissible trajectory $\boldsymbol{x}^*$ (according to Equation 1.71b) that minimize J. Note that an optimal control problem may be seen as a generalization of functional optimization under constraints.

However, to set the problem in a proper format of constrained optimization, the first term in Equation (1.71a) on the right hand side (RHS) is written hereunder as:

$$h\left(\boldsymbol{x}\left(t_f\right),t_f\right) = \int_{t_0}^{t_f} \frac{d}{dt}\left\{h\left(\boldsymbol{x}(t),t\right)\right\}dt + h\left(\boldsymbol{x}\left(t_0\right),t_0\right) \tag{1.72}$$

It is assumed that h is differentiable. Since $h\left(\boldsymbol{x}\left(t_0\right),t_0\right)$ is a scalar constant and the task of minimization is unaffected by this quantity, the performance index may be expressed as:

$$J = \int_{t_0}^{t_f} \left\{ L\left(\boldsymbol{x}(t),\boldsymbol{u}(t),t\right) + \frac{dh\left(\boldsymbol{x}(t),t\right)}{dt} \right\} dt$$

$$\Rightarrow J = \int_{t_0}^{t_f} \left\{ L\left(\boldsymbol{x}(t),\boldsymbol{u}(t),t\right) + \frac{\partial h\left(\boldsymbol{x}(t),t\right)}{\partial t} + \frac{\partial h\left(\boldsymbol{x}(t),t\right)}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} \right\} dt \tag{1.73}$$

The second step in the last equation is obtained by invoking the total derivative $\frac{d}{dt} = \frac{\partial}{\partial t} + \frac{\partial}{\partial \boldsymbol{x}}\frac{d\boldsymbol{x}}{dt}$. Now, by Lagrange multiplier method (see Equation 1.63), the optimization problem is recast as an unconstrained one as:

$$\text{minimize}\,\bar{J}\left(\boldsymbol{x},\dot{\boldsymbol{x}},\boldsymbol{u},\boldsymbol{p},t\right)=\int_{t_0}^{t_f}\left\{\begin{array}{c}L\left(\boldsymbol{x}(t),\boldsymbol{u}(t),t\right)+\dfrac{\partial h\left(\boldsymbol{x}(t),t\right)}{\partial t}+\dfrac{\partial h\left(\boldsymbol{x}(t),t\right)}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}}\\ +\boldsymbol{p}^T\left(\boldsymbol{F}\left(\boldsymbol{x}(t),\boldsymbol{u}(t),t\right)-\dot{\boldsymbol{x}}\right)\end{array}\right\}dt \tag{1.74}$$

where $\boldsymbol{p}(t)=\left\{p_i(t),i=1,2,\ldots,n\right\}$ are the time-varying Lagrange multipliers associated with the differential equation (DE) constraints in Equation (1.71b). Note that the performance index as posed in the last equation is similar to an action integral as is known in calculus of variations.

Pontryagin's minimum (or maximum) principle

The well-known Pontryagin minimum (or maximum) principle (Pontryagin et al., 1962; Stengel, 1994; Kirk 1970) provides the necessary conditions for optimality and it is indeed a generalisation of the classical subject of the calculus of variations to optimal control theory. The vector of terminal condition $\boldsymbol{x}\left(t_f\right)$ may be specified or otherwise (free). The first variation $\delta\bar{J}$ in terms of the variations $\delta\boldsymbol{x},\delta\dot{\boldsymbol{x}},\delta\boldsymbol{p},\delta\boldsymbol{u}$ and δt_f is obtained by first writing an increment $\Delta\bar{J}$ as:

$$\Delta\bar{J}\left(\boldsymbol{x},\dot{\boldsymbol{x}},\boldsymbol{u},\boldsymbol{p},t\right)=\bar{J}\left(\boldsymbol{x}+\delta\boldsymbol{x}_f,\dot{\boldsymbol{x}}+\delta\dot{\boldsymbol{x}}_f,\boldsymbol{u}+\delta\boldsymbol{u},\boldsymbol{p}+\delta\boldsymbol{p},\delta t_f+\delta\delta t_f\right)-\bar{J}\left(\boldsymbol{x},\dot{\boldsymbol{x}},\boldsymbol{u},\boldsymbol{p},t\right) \tag{1.75}$$

Expanding the expression on the RHS over the variations, retaining only the linear terms and integration by parts gives (Kirk 1970):

$$\delta\bar{J}\left(\boldsymbol{x},\dot{\boldsymbol{x}},\boldsymbol{u},\boldsymbol{p},t\right)=\frac{\partial\bar{L}\left(t_f\right)}{\partial\dot{\boldsymbol{x}}}^T\delta\boldsymbol{x}_f+\left\{\bar{L}\left(t_f\right)-\frac{\partial\bar{L}\left(t_f\right)}{\partial\dot{\boldsymbol{x}}}^T\dot{\boldsymbol{x}}^*\left(t_f\right)\right\}\delta t_f$$

$$+\int_{t_0}^{t_f}\left\{\left(\frac{\partial\bar{L}(t)}{\partial\boldsymbol{x}}^T-\frac{d}{dt}\frac{\partial\bar{L}(t)}{\partial\dot{\boldsymbol{x}}}^T\right)\delta\boldsymbol{x}(t)+\frac{\partial\bar{L}(t)}{\partial\boldsymbol{u}}^T\delta\boldsymbol{u}(t)+\frac{\partial\bar{L}(t)}{\partial\boldsymbol{p}}^T\delta\boldsymbol{p}(t)\right\}dt \tag{1.76a}$$

where $\bar{L}(t)$ is given by (see Equation 1.74):

$$\bar{L}(t)=L\left(\boldsymbol{x}(t),\boldsymbol{u}(t),t\right)+\frac{\partial h\left(\boldsymbol{x}(t),t\right)}{\partial t}+\frac{\partial h\left(\boldsymbol{x}(t),t\right)}{\partial\boldsymbol{x}}\dot{\boldsymbol{x}}+\boldsymbol{p}^T\left(\boldsymbol{F}\left(\boldsymbol{x}(t),\boldsymbol{u}(t),t\right)-\dot{\boldsymbol{x}}\right) \tag{1.76b}$$

In Equation (1.76a), for the terms inside the integral involving $h\left(\boldsymbol{x}(t),t\right)$ in $\bar{L}(t)$, one finds that $\dfrac{\partial}{\partial\boldsymbol{u}}\left(\dfrac{\partial h\left(\boldsymbol{x}(t),t\right)}{\partial t}\right)=0$ and $\dfrac{\partial}{\partial\boldsymbol{p}}\left(\dfrac{\partial h\left(\boldsymbol{x}(t),t\right)}{dt}\right)=0$. Also, the coefficient terms of $\delta\boldsymbol{x}(t)$ in the integral containing $h\left(\boldsymbol{x}(t),t\right)$ sum up to:

$$\frac{\partial}{\partial \boldsymbol{x}}\left(\frac{\partial h\left(\boldsymbol{x}(t),t\right)}{\partial \boldsymbol{x}}^{T}\dot{\boldsymbol{x}}+\frac{\partial h\left(\boldsymbol{x}(t),t\right)}{\partial t}\right)-\frac{d}{dt}\left(\frac{\partial}{\partial \dot{\boldsymbol{x}}}\left(\frac{\partial h\left(\boldsymbol{x}(t),t\right)}{\partial \boldsymbol{x}}^{T}\dot{\boldsymbol{x}}\right)\right)$$

$$=\frac{\partial}{\partial \boldsymbol{x}}\left(\frac{\partial h\left(\boldsymbol{x}(t),t\right)}{\partial \boldsymbol{x}}^{T}\dot{\boldsymbol{x}}+\frac{\partial h\left(\boldsymbol{x}(t),t\right)}{\partial t}\right)-\frac{d}{dt}\left(\frac{\partial h\left(\boldsymbol{x}(t),t\right)}{\partial \boldsymbol{x}}\right) \tag{1.77}$$

which reduces to zero after applying chain rule to $\frac{d}{dt}\left(\frac{\partial h\left(\boldsymbol{x}(t),t\right)}{\partial \boldsymbol{x}}\right)$ and by interchanging differentiations with respect to $\boldsymbol{x}$ and t. With this simplification, equating the first variation $\delta\bar{J}$ to zero and noting the arbitrariness of the variations involved gives the optimality conditions as:

$$\text{coefficient of } \delta\boldsymbol{p}(t):\frac{\partial \bar{L}(t)}{\partial \boldsymbol{p}}=0\Rightarrow \dot{\boldsymbol{x}}^{*}(t)=\boldsymbol{F}\left(\boldsymbol{x}^{*}(t),\boldsymbol{u}^{*}(t),t\right) \tag{1.78a}$$

$$\text{Coefficient of } \delta\boldsymbol{x}(t):\frac{\partial \bar{L}(t)}{\partial \boldsymbol{x}}^{T}-\frac{d}{dt}\frac{\partial \bar{L}(t)}{\partial \dot{\boldsymbol{x}}}^{T}=0\Rightarrow -\dot{\boldsymbol{p}}^{*}=\frac{\partial L\left(\boldsymbol{x}^{*}(t),\boldsymbol{u}^{*}(t),t\right)}{\partial \boldsymbol{x}}+\left[\frac{\partial \boldsymbol{F}\left(\boldsymbol{x}^{*}(t),\boldsymbol{u}^{*}(t),t\right)}{\partial \boldsymbol{x}}\right]^{T}\boldsymbol{p}^{*}(t) \tag{1.78b}$$

$$\text{coefficient of } \delta\boldsymbol{u}(t):\frac{\partial \bar{L}(t)}{\partial \boldsymbol{u}}^{T}=0\Rightarrow \frac{\partial L\left(\boldsymbol{x}^{*}(t),\boldsymbol{u}^{*}(t),t\right)}{\partial \boldsymbol{u}}+\left[\frac{\partial \boldsymbol{F}\left(\boldsymbol{x}^{*}(t),\boldsymbol{u}^{*}(t),t\right)}{\partial \boldsymbol{u}}\right]^{T}\boldsymbol{p}^{*}(t)=0 \tag{1.78c}$$

The terms outside the integral in Equation (1.76a) yield the natural BCs:

$$\frac{\partial \bar{L}\left(t_f\right)}{\partial \dot{\boldsymbol{x}}}^{T}\delta\boldsymbol{x}_f+\left\{\bar{L}\left(t_f\right)-\frac{\partial \bar{L}\left(t_f\right)}{\partial \dot{\boldsymbol{x}}}^{T}\dot{\boldsymbol{x}}^{*}\left(t_f\right)\right\}\delta t_f=0$$

$$\Rightarrow\left\{\frac{\partial h\left(\boldsymbol{x}^{*}\left(t_f\right),t_f\right)}{\partial \boldsymbol{x}}-\boldsymbol{p}^{*}\left(t_f\right)\right\}^{T}\delta\boldsymbol{x}_f+\left\{\begin{array}{c}L\left(\boldsymbol{x}^{*}\left(t_f\right),\boldsymbol{u}^{*}\left(t_f\right),t_f\right)+\frac{\partial h\left(\boldsymbol{x}^{*}\left(t_f\right),t_f\right)}{\partial t}\\ +\boldsymbol{p}^{*T}\left(t_f\right)F\left(\boldsymbol{x}^{*}\left(t_f\right),\boldsymbol{u}^{*}\left(t_f\right),t_f\right)\end{array}\right\}\delta t_f=0 \tag{1.79}$$

Equation (1.78a) represents the state equations satisfied by the optimal control $\boldsymbol{u}^*(\boldsymbol{t})$ and optimal trajectory $\boldsymbol{x}^*(t)$. The n-dimensional vector $\boldsymbol{p}(t)$ of Lagrange multipliers is known as the costate vector and Equations (1.78b) the costate equations. Equation (1.78c) is a set of m algebraic equations to be satisfied by $\boldsymbol{u}^*(\boldsymbol{t})$ and $\boldsymbol{x}^*(t)$ for all $t \in \left[t_0, t_f\right]$. Solution of the state and costate DEs require $2n$ constants of integration supplied by n initial conditions $\boldsymbol{x}^*(0) = \boldsymbol{x}_0$ and n BCs in Equation (1.79).

Now, consider the function:

$$H\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}(t), t\right) = L\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) + \boldsymbol{p}^T(t)\boldsymbol{F}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) \tag{1.80}$$

Equations (1.78a-c) may be restated in terms of H as:

$$\dot{\boldsymbol{x}}^*(t) = \frac{\partial H\left(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{p}^*(t), t\right)}{\partial \boldsymbol{p}} \tag{1.81a}$$

$$\dot{\boldsymbol{p}}^*(t) = -\frac{\partial H\left(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{p}^*(t), t\right)}{\partial \boldsymbol{x}} \tag{1.81b}$$

and

$$\frac{\partial H\left(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{p}^*(t)t\right)}{\partial \boldsymbol{u}} = 0 \tag{1.81c}$$

The BCs in Equation (1.79) take the form:

$$\left\{\frac{\partial h\left(\boldsymbol{x}^*\left(t_f\right), t_f\right)}{\partial \boldsymbol{x}} - \boldsymbol{p}^*\left(t_f\right)\right\}^T \delta \boldsymbol{x}_f + \left(H\left(\boldsymbol{x}^*\left(t_f\right), \boldsymbol{u}^*\left(t_f\right), \boldsymbol{p}^*\left(t_f\right), t_f\right) + \frac{\partial h\left(\boldsymbol{x}^*\left(t_f\right), t_f\right)}{\partial t}\right)\delta t_f = 0 \tag{1.82}$$

Equations (1.81a-b) resemble the canonical Hamiltonian equations which appear in classical mechanics (also see Meirovitch 1990). The Hamiltonian and Lagrangian are related by the Legendre transform (Appendix 1). Presently, the costate variables $\boldsymbol{p}(t)$ bear no similarity to the generalized momenta $m\boldsymbol{v}(= m\dot{\boldsymbol{x}})$ usually implied in the Hamiltonian equations in mechanics. In optimal control theory, $H\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}(t), t\right)$ is called a pseudo-Hamiltonian function. Equation (1.81c) is a necessary condition for optimality when the control variable $\boldsymbol{u}(t)$ is unconstrained.

The control variables $\boldsymbol{u}(t)$ are generally bounded which is often the case from practical considerations. Suppose that the admissible controls are constrained to a set

U, i.e. $\boldsymbol{u}(t) \in U$. Considering the effect of control variables on a functional $J(\boldsymbol{u})$, the incremental quantity $\Delta J(\boldsymbol{u}^*, \delta\boldsymbol{u})$ for any $\boldsymbol{u}$ and optimal $\boldsymbol{u}^*$ is given by:

$$\Delta J(\boldsymbol{u}^*, \delta\boldsymbol{u}) = J(\boldsymbol{u}) - J(\boldsymbol{u}^*) = \delta J(\boldsymbol{u}^*) + \text{higher order terms of } o(\delta\boldsymbol{u}) \quad (1.83)$$

where $\boldsymbol{u} = \boldsymbol{u}^* + \delta\boldsymbol{u}$. $\delta J(\boldsymbol{u}^*)$ is the first variation, which is linear in $\delta\boldsymbol{u}$.

The functional $J(\boldsymbol{u})$ has a relative minimum at $\boldsymbol{u}^*$ if $\Delta J(\boldsymbol{u}^*, \delta\boldsymbol{u}) \geq 0$. For unbounded control input, $\delta\boldsymbol{u}$ is assumed to be arbitrary, leading to the condition that the first variation $\delta J(\boldsymbol{u}^*) = 0$ for all admissible $\delta\boldsymbol{u}$ having a sufficiently small norm, i.e. $\delta\boldsymbol{u} := \int_{t_0}^{t_f} \sum_{i=1}^{m} |\delta u_i(t)| dt < \alpha$, say. This argument was in fact used in deriving the optimal control and trajectory in the last section (Equations 1.78 or 1.79) with $\delta J(\boldsymbol{u}^*) = 0$. Two cases may however arise for bounded inputs. One is that the optimal $\boldsymbol{u}^*(t)$ lies strictly within the specified boundary $\forall t \in [t_0, t_f]$ so that the assumption of $\delta\boldsymbol{u}$ being arbitrary remains valid. It is similar to the case of unbounded inputs. Alternatively, with a partition of the interval $[t_0, t_f]$ expressed by $t_0 < t_1 < \ldots < t_i < \ldots < t_f$, if the variation $\delta\boldsymbol{u}$ happens to cross over to the inadmissible region over an interval (t_i, t_{i+1}) as shown in Figure 1.28, such variations are not admissible. In this case, the optimality condition for a minimum is $\delta J(\boldsymbol{u}^*) > 0$.

With the above arguments on admissible variations, we now consider the pseudo-Hamiltonian H in Equation (1.80) corresponding to the functional $\bar{J}$ in Equation

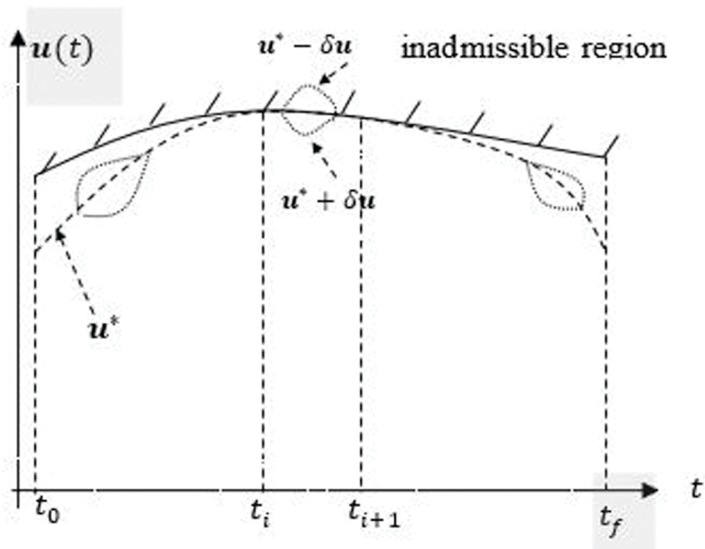


FIGURE 1.28 Optimal control problem; u^* lying on the boundary and control input variation δu outside the admissible region in some interval $(t_i, t_{i+1}) \in [t_0, t_f]$.

(1.74). The optimality conditions with respect to $\delta \boldsymbol{p}$ and $\delta \boldsymbol{x}$ remain the same as in Equations (1.81a) and (1.81b). The BC in Equation (1.82) is also assumed to hold. Thus, with the coefficients of $\delta \boldsymbol{x}$ and $\delta \boldsymbol{p}$ in $\delta \bar{J}$ identically zero, it remains only to seek optimality with respect to the variation $\delta \boldsymbol{u}(t)$, i.e.:

$$\int_{t_0}^{t_f} \frac{\partial H\left(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{p}^*(t), t\right)^T}{\partial \boldsymbol{u}} \delta \boldsymbol{u}(t)\, dt \geq 0 \tag{1.84}$$

$$\Rightarrow \int_{t_0}^{t_f} \left(H\left(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t) + \delta \boldsymbol{u}(t), \boldsymbol{p}^*(t), t\right) - H\left(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{p}^*(t), t\right)\right) dt \geq 0 \tag{1.85}$$

Thus, for all admissible $\delta \boldsymbol{u}$ and $\forall t \in \left[t_0, t_f\right]$, we must have:

$$H\left(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t) + \delta \boldsymbol{u}(t), \boldsymbol{p}^*(t), t\right) - H\left(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{p}^*(t), t\right) \geq 0 \tag{1.86}$$

following which we get the necessary condition for $\boldsymbol{u}^*$ to minimize $\bar{J}\ \forall t \in \left[t_0, t_f\right]$ as:

$$H\left(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{p}^*(t), t\right) \leq H\left(\boldsymbol{x}^*(t), \boldsymbol{u}(t), \boldsymbol{p}^*(t), t\right) \tag{1.87}$$

Strict inequality in the above equation implies that the optimal control $\boldsymbol{u}^*(t)$ minimizes H under bounded $\boldsymbol{u}(t)$ and this is Pontryagin's minimum principle. Thus, for bounded controls, Equation (1.87) replaces Equation (1.81c). This condition along with Equations (1.81a) and (1.81b) and supplemented by the BC in Equation (1.82) constitute the necessary conditions for the n-dimensional optimal control problem under bounded inputs. Pontryagin's minimum principle equally applies to the case of unbounded controls. This is because, as bounds on the admissible controls tend to infinity, the region is unbounded as well. It amounts to an unconstrained optimization problem with the optimality condition given by Equation (1.81c).

Example 1.5. We derive the optimal control law for an n-dimensional dynamical system described by:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{1.88a}$$

when the performance index to be minimized is:

$$J\left(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{u}, t\right) = \frac{1}{2}\boldsymbol{x}^T\left(t_f\right)\boldsymbol{S}\boldsymbol{x}\left(t_f\right) + \frac{1}{2}\int_{t_0}^{t_f}\left[\boldsymbol{x}^T(t)\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}\boldsymbol{u}(t)\right]dt \tag{1.88b}$$

Here $\boldsymbol{x}(t)=\left(x_1(t), x_2(t), x_3(t), x_4(t)\right)^T$. $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, $\boldsymbol{B}\in\mathbb{R}^{n\times m}$, $\boldsymbol{x}\in\mathbb{R}^n$, $\boldsymbol{u}\in\mathbb{R}^m$; $\boldsymbol{S}$ and $\mathbf{Q}$ are $n\times n$ real symmetric positive semidefinite constant matrices and $\boldsymbol{R}$ is an $m\times m$ real symmetric positive definite constant matrix. Assume that the controls are unbounded and that $\boldsymbol{x}\left(t_f\right)$ is free and t_f fixed.

Solution. This is known as a linear quadratic regulator problem. The task is to drive the system to a constant final state $\boldsymbol{x}\left(t_f\right)$ starting from a specified initial state. By an appropriate coordinate transformation, the task is to design a control input $\boldsymbol{u}^*(t)$ so as to bring the system to a zero state at t_f. The Hamiltonian is:

$$H\left(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{p}(t),t\right)=\frac{1}{2}\left[\boldsymbol{x}^T(t)\boldsymbol{Q}\boldsymbol{x}(t)+\boldsymbol{u}^T(t)\boldsymbol{R}\boldsymbol{u}(t)\right]+\boldsymbol{p}^T(t)\left[\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)\right] \tag{1.89}$$

According to Equation (1.88), the optimality conditions are:

$$\dot{\boldsymbol{x}}^*(t)=\frac{\partial H\left(\boldsymbol{x}^*(t),\boldsymbol{u}^*(t),\boldsymbol{p}^*(t),t\right)}{\partial \boldsymbol{p}}=\boldsymbol{A}\boldsymbol{x}^*(t)+\boldsymbol{B}\boldsymbol{u}^*(t) \tag{1.90a}$$

$$\dot{\boldsymbol{p}}^*(t)=-\frac{\partial H\left(\boldsymbol{x}^*(t),\boldsymbol{u}^*(t),\boldsymbol{p}^*(t),t\right)}{\partial \boldsymbol{x}}=-\left\{\boldsymbol{Q}\boldsymbol{x}^*(t)+\boldsymbol{A}^T\boldsymbol{p}^*(t)\right\} \tag{1.90b}$$

and

$$\frac{\partial H\left(\boldsymbol{x}^*(t),\boldsymbol{u}^*(t),\boldsymbol{p}^*(t)t\right)}{\partial \boldsymbol{u}}=0=\boldsymbol{R}\boldsymbol{u}^*(t)+\boldsymbol{B}^T\boldsymbol{p}^*(t) \tag{1.90c}$$

Equation (1.79) gives the BC:

$$\boldsymbol{p}^*\left(t_f\right)=\boldsymbol{S}\boldsymbol{x}\left(t_f\right) \tag{1.91}$$

From Equation (1.90c),

$$\boldsymbol{u}^*(t)=-\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{p}^*(t) \tag{1.92}$$

We assume that the costate vector $\boldsymbol{p}^*(t)$ is linearly related to $\boldsymbol{x}^*(\boldsymbol{t})$ as:

$$\boldsymbol{p}^*(t)=\boldsymbol{K}(t)\boldsymbol{x}^*(t) \tag{1.93}$$

$\boldsymbol{K}(t)$ is an $n\times n$ matrix. With this assumption, Equations (1.90a-b), (1.92) and (1.93) yield:

$$\dot{\boldsymbol{K}}(t)\boldsymbol{x}^*(t)+\boldsymbol{K}(t)\boldsymbol{A}\boldsymbol{x}^*(t)-\boldsymbol{K}(t)\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{K}(t)\boldsymbol{x}^*(t)=-\boldsymbol{Q}\boldsymbol{x}^*(t)-\boldsymbol{A}^T\boldsymbol{K}(t)\boldsymbol{x}^*(t)$$

$$\Rightarrow \dot{\boldsymbol{K}}(t)=-\boldsymbol{Q}-\boldsymbol{A}^T\boldsymbol{K}(t)-\boldsymbol{K}(t)\boldsymbol{A}+\boldsymbol{K}(t)\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{K}(t) \tag{1.94}$$

From Equation (1.91),

$$\boldsymbol{K}(t_f)=\boldsymbol{S} \tag{1.95}$$

The matrix DE (1.94) is known as the Riccati equation and consists of n^2 nonlinear DEs. Since $\boldsymbol{K}^T(t)$ also satisfies the Riccati equation, $\boldsymbol{K}(t)$ is a symmetric matrix and it suffices to solve the $n(n+1)/2$ DEs. Using the boundary condition in Equation (1.95), the Riccati equation can be solved backwards from t_f to t_0. Once $\boldsymbol{K}(t)$ – the Riccati matrix, is determined, we obtain the control $\boldsymbol{u}^*(t)=-\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{p}^*(t)=-\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{K}(t)\boldsymbol{x}^*(t)=\boldsymbol{G}(t)\boldsymbol{x}(t)$ where $\boldsymbol{G}(t)$ is the feedback gain matrix (Figure 1.29).

Consider a specific example with $n=4$ and $m=1$ and with the state equation given by:

$$\dot{\boldsymbol{x}}(t)=A\boldsymbol{x}+\boldsymbol{B}u(t) \tag{1.96}$$

where $\boldsymbol{x}(t)=\{x_1(t),x_2(t),x_3(t),x_4(t)\}^T$ and

$$\boldsymbol{A}=\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -100 & 100 & 0 & 0\\ 100 & -200 & 0 & 0\end{bmatrix},\boldsymbol{B}=\begin{bmatrix}0\\0\\0\\1\end{bmatrix} \tag{1.97}$$

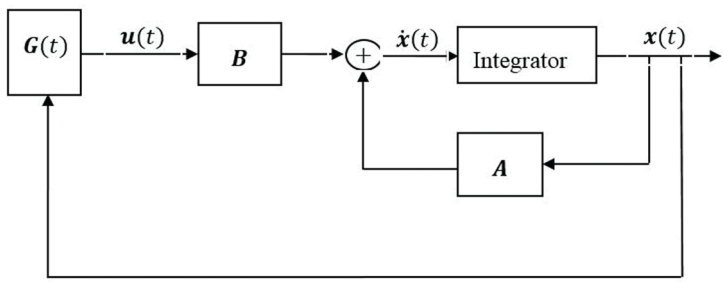


FIGURE 1.29 Optimal control; linear regulator problem and feedback control.

Let $\boldsymbol{Q} = \boldsymbol{I}$, a 4×4 identity matrix and $R = 1$.The Hamiltonian is:

$$H\left(\boldsymbol{x}(t), u(t), \boldsymbol{p}(t), t\right) = \frac{1}{2}\left[\boldsymbol{x}^T(t)\boldsymbol{Q}\boldsymbol{x}(t) + u^2(t)\right] + \boldsymbol{p}^T(t)\left[\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u(t)\right] \quad (1.98)$$

and the Riccati matrix $\boldsymbol{K}(t)$ in Equation (1.94) need to be solved with the boundary condition $\boldsymbol{K}\left(t_f\right) = \boldsymbol{S} = \boldsymbol{0}$ to arrive at the optimal control:

$$u^*(t) = -R^{-1}\boldsymbol{B}^T\boldsymbol{p}^*(t) = -\boldsymbol{B}^T\boldsymbol{K}(t)\boldsymbol{x}^*(t) = -\boldsymbol{G}(t)\boldsymbol{x}^*(t) \quad (1.99)$$

An elegant method (Meirovitch 1990) to solve for $\boldsymbol{K}(t)$ is by a matrix transformation:

$$\boldsymbol{K}(t) = \mathbb{E}(t)\mathbb{F}^{-1}(t) \quad (1.100)$$

$\mathbb{E}$, $\mathbb{F} \in \mathbb{R}^{n \times n}$. The transformation helps in converting the set of n^2 nonlinear ODEs in Equation (1.94) to a set of $2n^2$ linear ODEs:

$$\begin{Bmatrix} \dot{\mathbb{E}}(t) \\ \dot{\mathbb{F}}(t) \end{Bmatrix} = \begin{bmatrix} -\boldsymbol{A}^T & -Q \\ -BR^{-1}\boldsymbol{B}^T & A \end{bmatrix} \begin{Bmatrix} E(t) \\ F(t) \end{Bmatrix} \quad (1.101)$$

The advantage of having linear ODES to be solved often outweighs the extra effort needed to handle twice the number of equations. In the present example, $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{Q}$ are constant matrices. In general, these may be time varying matrices. The BC in Equation (1.95) may be taken as $\boldsymbol{K}\left(t_f\right) = \boldsymbol{S} = \mathbb{E}\left(t_f\right)\mathbb{F}^{-1}\left(t_f\right)$ from which the BCs for $\mathbb{E}(t)$ and $\mathbb{F}(t)$ may be assumed as:

$$\mathbb{E}\left(t_f\right) = \boldsymbol{S}, \mathbb{F}(t_f) = \boldsymbol{I} \quad (1.102)$$

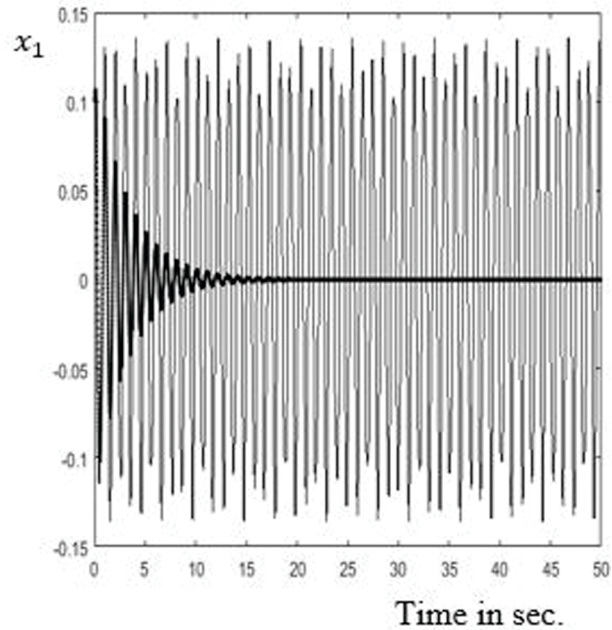
Equation (1.101) is solved backwards with $t_f = 50\,\text{sec.}$ and with the help of the BCs in the last equation, to obtain the Riccati matrix $\boldsymbol{K}(t)$. With the feedback gain matrix $\boldsymbol{G}(t) = \boldsymbol{B}^T\boldsymbol{K}(t)$, the system equation to be solved under optimal control $u^*(t) = -\boldsymbol{G}(t)\boldsymbol{x}^*(t)$ is:

$$\dot{\boldsymbol{x}}^*(t) = \left[\boldsymbol{A} - \boldsymbol{B}\boldsymbol{G}\right]\boldsymbol{x}^*(t) \quad (1.103)$$

Figure 1.30 shows the solution to Equation (1.103) under the initial conditions $\boldsymbol{x}_i^*\left(t_0\right) = 0.1, i = 1,2,3,4$. The system dynamics with and without optimal control are shown in the figure.

■

The continuous-time optimal control problem may also be solved via the Bellman principle of optimality (Bellman and Kalaba 1964). The principle leads to the



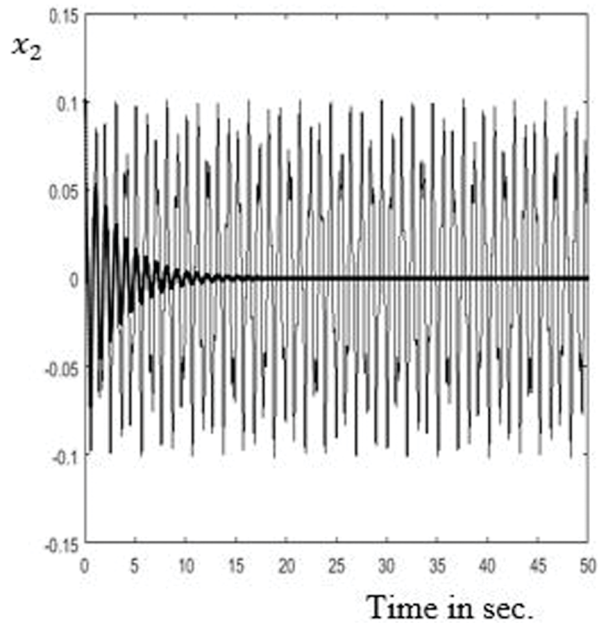
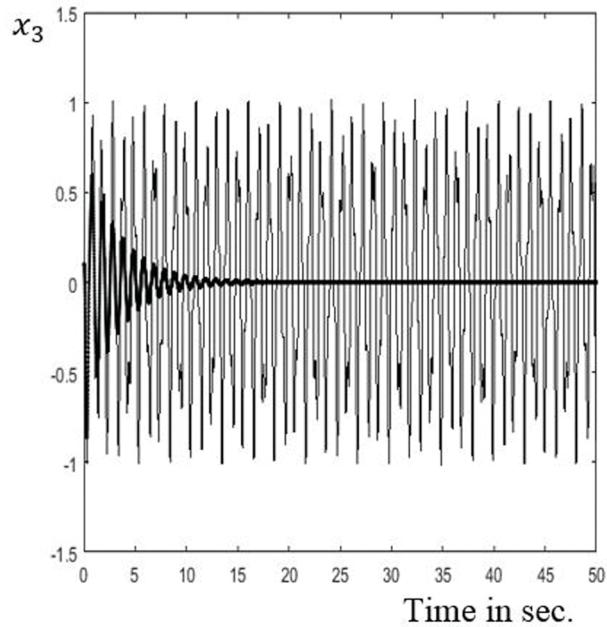


FIGURE 1.30a–b LQR problem; system state optimal trajectories along with the uncontrolled ones: (a) $x_1(t)$, (b) $x_2(t)$, (c) $x_3(t)$ and (d) $x_4(t)$, light black – uncontrolled, dark black – controlled.



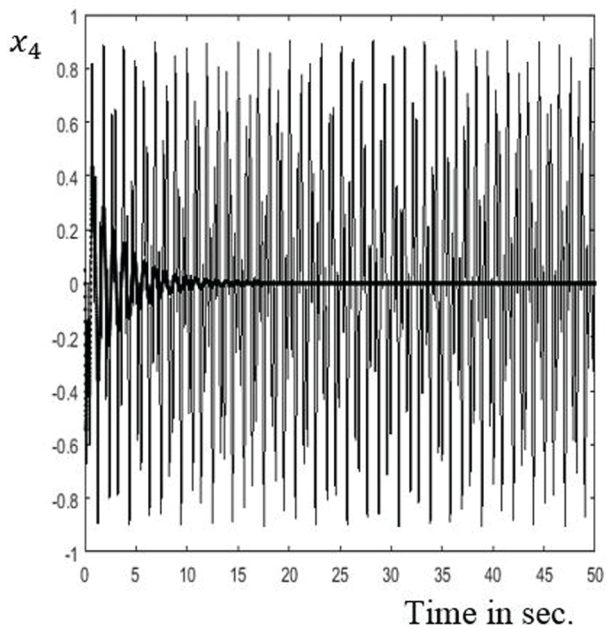


FIGURE 1.30c–d (Continued)

well-known Hamilton-Jacobi-Bellman (HJB) equation which is a PDE; see Appendix 1 for some details. For an LQR problem, the solution to the control variable $\boldsymbol{u}(t)$ in terms of $\boldsymbol{x}(t)$ leads to the same Riccati Equation (1.94) by the Bellman principle of optimality.

CONCLUDING REMARKS

In this introductory chapter, we have discussed the role of optimization in various fields whilst taking a bird's eye view of how to pose these problems and derive appropriate optimality conditions. The reader may thus have had a feel of the all-encompassing presence of optimization problems in applied science and engineering. The interesting origins of optimization theory are discussed particularly with reference to the famous traveling salesman and brachistochrone problems. Solutions to the traveling salesman problem are illustrated, first by a simple heuristic (brute force) method and then using the Metropolis algorithm. The close link the Metropolis algorithm has with probability theory is highlighted. In this context, definitions of local and global optima are also provided. It is often the local optimum rather than the global that may be possible to realize in complex optimization problems. The necessary and sufficient optimality conditions for both unconstrained and constrained optimization problems are discussed. An intuitive understanding of the optimality conditions is provided from geometric considerations too.

As we have noted, the brachistochrone problem and its solution date back to the late 17th century that marked the emergence of 'calculus of variations', which is indeed the basic theory of functional optimization. This subsequently paved the way to solutions to myriad other problems in science and engineering, such as optimal control and systems analysis using the finite element method. In order to motivate our readership, a brief account of these applications is also presented in the chapter.

Given the wide and cross-disciplinary interest, many optimization schemes – some though not all of which are rigorously grounded, had emerged in the last century. Owing to a sound mathematical basis, classical methods such as the ones based on derivatives (gradient and penalty techniques) and others that are derivative-free (pattern search and simplex methods, for instance) will be of special interest in this introductory text. Chapter 2 is devoted to derivative-based methods.

NOTATIONS – CHAPTER 1

A	area of cross-section
$\boldsymbol{A}$	coefficient matrix in the state space equation (Equation 1.88a)
$\mathcal{A}$	differential operator (Equation 1.42)
$\boldsymbol{B}$	coefficient matrix in the state space equation (Equation 1.88a)
$\mathcal{B}_r(\boldsymbol{x}_0)$	open ball with centre $\boldsymbol{x}_0$ and radius $r > 0$ in $\mathbb{R}^n$
c	speed of light in vacuum
c_1, c_2	real constants

C	real constant
$\boldsymbol{C}$	covariance matrix
$\boldsymbol{d}$	direction vector (Equation 1.46)
$d(\boldsymbol{x},\boldsymbol{y})$	distance function (metric) between points $\boldsymbol{x}$ and $\boldsymbol{y}$
$\mathcal{D}$	feasible space
E	Young's modulus of elasticity
$\boldsymbol{E}$	vector of edges in a complete graph (Figure 1.3)
$\mathbb{E}(t)$	time dependent matrix in Equation (1.100)
E_{ij}	edges in the graph of Figure 1.3
$f(\boldsymbol{x})$	objective function
$f_A(x,t)$	axial force density per unit length (Figure 1.19)
F	cone of descent directions
$F_i,\ i=0,1,\ldots$	Fibonacci numbers
$\mathbb{F}(t)$	time dependent matrix in Equation (1.100)
g	acceleration gravity (Equation 1.12)
$g(\boldsymbol{x}),\ \boldsymbol{g}(\boldsymbol{x})$	inequality constraints
$G=(\boldsymbol{V},\boldsymbol{E})$	complete graph ($\boldsymbol{V}$-vector of nodes and $\boldsymbol{E}$-vector of sides)
G	cone of feasibility directions for the inequality constraints
$h(x)$	an arbitrary function (Equation 1.13)
$\boldsymbol{h}(\boldsymbol{x})$	vector of equality constraints
$h(\boldsymbol{x},t)$	a scalar function : $\mathbb{R}^n\times\mathbb{R}^+\to R$ in Equation (1.71a)
H	set of tangent directions of equality constraints
$\mathcal{H}$	Hilbert space
$\boldsymbol{H}(\boldsymbol{x})$	Hessian matrix
$H(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{p}(t),t)$	Hamiltonian in Equation (1.80)
$I(y)$	Action integral
J	performance index in optimal control problem
K_B	Boltzmann constant
$\mathcal{K}(.,.)$	bilinear form (Equation 1.43)
$\boldsymbol{K}$	stiffness matrix (symmetric) – Equation (1.45)
$\boldsymbol{K}(t)$	time dependent symmetric matrix in Riccati Equation (1.94)
$\ell(.)$	Linear form (Equation 1.43)
L	Lagrangian
$L(\boldsymbol{x},\boldsymbol{u},t)$	a scalar function: $\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}^+\to R$ in Equation (1.71a)
ʟ	Lagrangian density
m	mass density
$\mathcal{M}(.,.)$	bilinear form (Equation 1.43)
$\boldsymbol{M}$	mass matrix (symmetric) (Equation 1.45)
N	an integer
N_d	number of nodes in a finite element discretization

$\boldsymbol{p}(t)=\{p_i(t), i=1,2,\ldots,n\}$	time-varying Lagrangian multipliers in optimal control problem
$\mathcal{P}=\mathcal{P}_j,\ j=1,2,\ldots,N_d$	external load at the nodes (Equation 1.45)
$q_j(t)$	generalized coordinates (Equation 1.41)
Q	symmetric positive semidefinite constant matrix
R	symmetric positive definite constant matrix
$\mathcal{R}(\tilde{y})=\mathcal{A}(\tilde{y})-f$	residual in Equation (1.42)
S	symmetric positive semidefinite constant matrix
t_f	final time
$T(.)$	Minimum time of descent (Equation 1.12)
T	kinetic energy of a system
$\mathcal{T}$	tangent subspace of the equality and active inequality constraints
$\boldsymbol{u}(t), u(t)$	control functions
$\mathcal{U}(\boldsymbol{x})$	utility (objective) function
$U_j(x)$	test functions (Equation 1.42)
v_1 and v_2	speeds of light in different media
ϑ	refractive index of a medium
V	potential energy of a system
V_i	cities in the graph of Figure 1.3
$\boldsymbol{V}$	vector of nodes in a complete graph (Figure 1.3)
w_{ij}	weights (Equation 1.8)
W_{nc}	work done by the non-conservative force (Equation 1.33)
$\boldsymbol{x}$	vector of design variables
$\boldsymbol{x}_L, \boldsymbol{x}_U$	lower and upper bounds for design variables
$\hat{\boldsymbol{x}}_k$	trial solution (in TSP)
$y(x)$	path in Figure 1.5
$\boldsymbol{Y}_j(\boldsymbol{x})$	trial functions (Equation 1.41)
Z_β	normalization constant (Equation 1.10)
δI	first variation of the action integral I in Equations (1.13) and (1.35)
$\delta y(x,t)$	virtual displacement over the true path $y(x,t)$
Ξ	space of design variables
θ_1, θ_2	angles of incidence and refraction (Equation 1.24)
$\boldsymbol{\lambda}=\lambda_i,\ i=1,2,\ldots$ and $\boldsymbol{\mu}=\mu_i,\ i=1,2,\ldots$	Lagrangian multipliers
$\Psi(\boldsymbol{x})$	neighborhood of $\boldsymbol{x}$, and $=\{\boldsymbol{y} \mid \boldsymbol{y}\in\mathcal{D}\cap d(\boldsymbol{x},\boldsymbol{y})\le\varepsilon\}$
$\nabla f(\boldsymbol{x})$	gradient of the objective function with respect to $\boldsymbol{x}$
$\nabla \boldsymbol{g}(\boldsymbol{x})$	gradient vector of (inequality) constraint functions with respect to $\boldsymbol{x}$
$\nabla \boldsymbol{h}(\boldsymbol{x})$	gradient vector of (equality) constraint functions with respect to $\boldsymbol{x}$
$\nabla_X L$	gradient of the Lagrangian L with respect to $\boldsymbol{X}$
$\nabla_{\lambda} L$	gradient of the Lagrangian L with respect to $\boldsymbol{\lambda}$

$\nabla_{\mu} L$	gradient of the Lagrangian L with respect to $\boldsymbol{\mu}$
$\nabla^2 L(\boldsymbol{x})$	Hessian of the Lagrangian L with respect $\boldsymbol{x}$
$\mho$	domain of interest (for FE semi-discretization)
$\{\mho_i\}, i = 1,2,\ldots,N_e$	non-overlapping elements in FE mesh

EXERCISES – CHAPTER 1

1. Find the optimum for the one-dimensional (unconstrained) optimization problem: minimize $f(x) = x^4 - 10x^3 + 20x + 2$ (Figure E1.1a) by using any of the conventional methods that reduce the interval of uncertainty containing the optimum point.

Notes: The popular methods that are used for the interval bracketing (Belegundu and Chandrupatla 1999) are (i) golden section method, (ii) Fibonacci search, (iii) quadratic fit and (iv) cubic fit.

Here a brief description of the *golden section method* as applicable to a minimization problem is provided below. Suppose that the initial interval that contains the optimum point x^* is given as (a_0, b_0) with $b_0 - a_0 = l_0 = l$. In golden search method, two intermediate points a_1 and b_1 are located within the given interval equidistant from the end points according to the section rule $a_1 = (1-\alpha) l_0$ and $b_1 = \alpha l_0$ with $\alpha < 1$ as shown in Figure E1.1b.

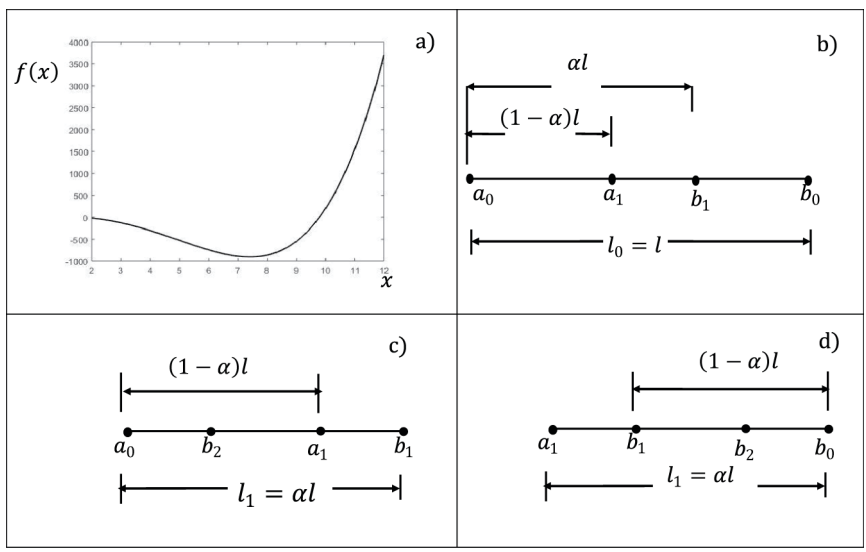


FIGURE E1.1 Golden section method: (a) unimodal function $f(x)$, (b) initial iteration and (c) next iteration if $f(a_1) < f(b_1)$, and (d) next iteration if $f(a_1) > f(b_1)$.

Case (A). If $f(a_1) < f(b_1)$, the interval (b_0, b_1) is rejected (Figure E1.1c). The bracketed interval is thus reduced to (a_0, b_1) whose length is $l_1 = \alpha l$. We proceed to the next iteration.

Case (B). On the other hand, if $f(a_1) > f(b_1)$, the interval (a_0, a_1) is rejected (Figure E1.1d) and the reduced interval is (a_1, b_0) whose length is again $l_1 = \alpha l$. We proceed to the next iteration.

Suppose that case (A) holds. With a_1 already fixed, a new point b_2 is to be located such that $b_1 - b_2 = a_1 - a_0 = (1-\alpha)l$. Thus $b_2 - a_0 = l_1 - (1-\alpha)l = (2\alpha - 1)l$. But by the section rule, $b_2 - a_0 = (1-\alpha)l_1 = (1-\alpha)\alpha l$. This leads to the condition $(2\alpha - 1)l = (1-\alpha)\alpha l$. To satisfy this condition, we must have $\alpha = \dfrac{-1 \pm \sqrt{5}}{2}$. Taking the positive value, we have $\alpha = 0.618$. This parameter α is known as the golden section ratio. If iterations are continued for 'n' times, one has the final reduced interval bracketing the optimum as $(0.618)^n l$. Suppose that it is required to have the final interval to be $l_\epsilon \ll l_0$, the number of iterations can be ascertained in advance from the equation $(0.618)^n l = l_\epsilon$.

The other alternative in case (B) is shown in Figure E1.1(d). In this case also, with b_1 already fixed, a new point b_2 is to be located such that $b_2 - a_1 = b_0 - b_1 = (1-\alpha)l$. The optimum point may be obtained as 7.4 at the end of 15 iterations with the specified $l_e = 0.01$. If the bisection method (with $\alpha = 0.5$ and only one intermediate point) is used, the optimum point is $\cong 7.625$.

2. Find the minimum of the function given in Exercise 1 by reducing the interval bracketing the optimum using Fibonacci method.

Notes: In golden section method (refer to the notes under the Exercise 1) the section ratio α is constant. In the Fibonacci method, it varies with iteration and may bracket the optimum point with fewer iterations. Here one uses the series of Fibonacci numbers $F_i, i = 1, 2, \ldots$ some of which are given in the table below.

i	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Fibonacci number, F_i	1	1	2	3	5	8	13	21	34

The sequence of Fibonacci numbers are generated as:

$$F_0 = 1, F_1 = 1 \text{ and } F_i = F_{i-1} + F_{i-2}, i = 2, 3, \ldots. \tag{E1.1}$$

Suppose that n is number of iterations (to be decided by the required accuracy). With l_0 being the given interval in which the optimum point is known to exist, let the starting interval for the first iteration be $l_1 = l_0$ and last interval l_n for the last iteration. We assume that the interval at any iteration follows the following reduction procedure.

$$l_1 = l_2 + l_3, l_2 = l_3 + l_4, \ldots, l_{n-2} = l_{n-1} + l_n \text{ and } l_{n-1} = 2l_n \tag{E1.2}$$

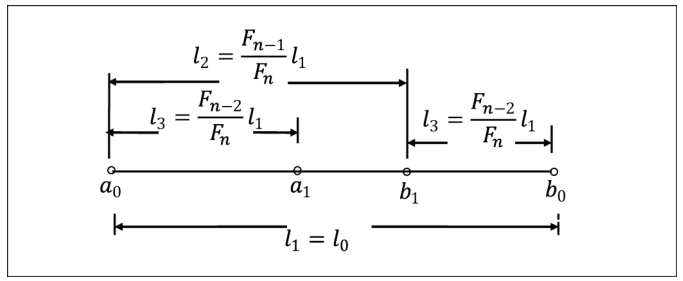


FIGURE E1.2 Fibanocci method, interval reduction – first iteration.

We can express the intermediate intervals in terms of last interval l_n as:

$$l_{n-1} = 2l_n, l_{n-2} = 3l_n, l_{n-3} = 5l_n, l_{n-4} = 8l_n, \ldots \tag{E1.3}$$

It is recognized that the coefficients are Fibonacci numbers, i.e., $l_{n-1} = F_2 l_n, l_{n-2} = F_3 l_n$, $l_{n-3} = F_4 l_n, l_{n-4} = F_5 l_n, \ldots$ and in general $l_{n-j} = F_{j+1} l_n$, $j = 1, 2, \ldots, n-1$. Thus for $j = n-1$, $l_1 = F_n l_n$. This suggests an iteration procedure based on the steps listed below.

Step 1. Fix the total number of iterations n based on the accuracy specified in terms of the final reduced interval length. That is, with a given l_n, we have $F_n = \frac{l_1}{l_n}$ and can fix n knowing F_n. For example, if $\frac{l_1}{l_n} = 20$, then $F_n = 20$ giving $n \approx 7$ (from the table of Fibonacci numbers).

Step 2. Comparing the function values at a_1 and b_1 (Figure E1.2), we can eliminate one of the intervals – either (a_0, a_1) or (b_0, b_1) - as discussed in the golden section method (see the notes under the Exercise 2) thus leading to a reduction in the interval from l_1 to $l_2 = (a_0, b_1)$ or (a_1, b_0).

Step 3. Repeating the same procedure of interval reduction, we finally arrive at the optimum point bracketed within the interval l_n.

For the given objective function in this exercise, the optimum point may be obtained approximately as 7.28 with specified $l_n = 0.01$ (which fixes $n = 16$ from the table of Fibanocci numbers).

3. For the one-dimensional (unconstrained) optimization problem in Exercise 1, find the minimum of the objective function $f(x) = x^4 - 10x^3 + 20x + 2$ by using the optimality condition $\frac{df}{dx} = 0$ and check that it is indeed the minimum from the nature of the Hessian matrix.

(Answer: $x^* = 7.4$)

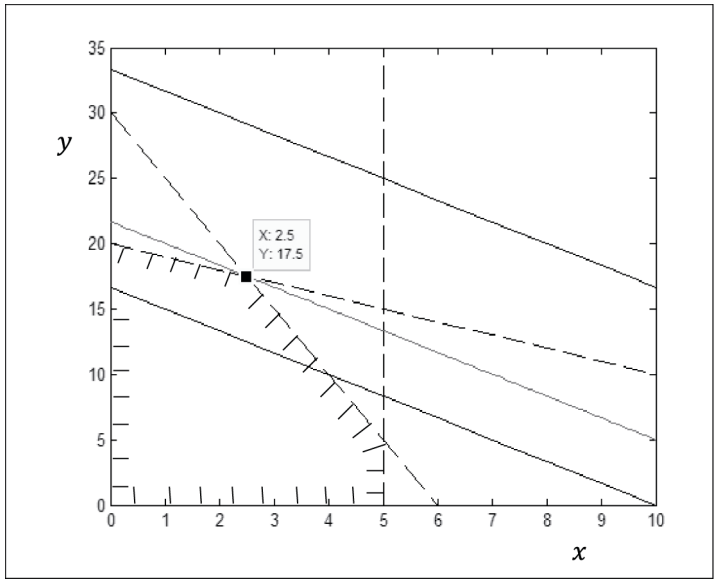


FIGURE E1.3 Maximization problem in Exercise 5, straight lines AB, CD and EF represent the limiting constraints, feasible region – shown in the first quadrant (hatched in the figure).

4. Discuss the polynomial based methods – quadratic fit and cubic fit – of finding the local minimum of one-dimensional unconstrained problems. Compare the two methods for the function $f(x) = \pi x^2 + \dfrac{1000}{x^2}$.

5. Consider the optimization problem defined by:

$$\text{maximize } f(\boldsymbol{x}) = 50x + 30y \tag{E1.4a}$$

s. t.

$$\begin{aligned} h_1(\boldsymbol{x}) = 5x + y - 30 \le 0, h_2(\boldsymbol{x}) = x + y - 20 \le 0, h_3(\boldsymbol{x}) \\ = x - 5 \le 0, h_4(\boldsymbol{x}) = x > 0 \text{ and } h_5(\boldsymbol{x}) = y > 0 \end{aligned} \tag{E1.4b}$$

Here $\boldsymbol{x} = (x, y)^T$. This is a linear programming (LP) problem with the objective function and the constraints being linear. Find the feasible region and find the optimum point by graphical construction.

Hint: The feasible region is shown in Figure E1.3. The optimum point is graphically obtained as $\boldsymbol{x}^* = (2.5, 17.5)^T$ - the graphical solution is also portrayed below.

6. Check the necessary and sufficiency condition for the optimization problem:

$$\text{Minimize } f(\boldsymbol{x}) = x_1 + x_2$$

$$\text{s.t. } x_1^2 + x_2^2 = 2 \qquad \text{(E1.5a,b)}$$

7. Given a set of observations $z_i, i = 1,2,\ldots,N$ in a random experiment, it is required to fit a probability density function (*pdf*). One assumes that each z_i is a realization sampled from *pdf*s of independent and identical random variables $Z_i, i = 1,2,\ldots,N$ (see Appendix 1 for details on random variables and their characterization). Assume that the *pdf* $\mathbb{f}_{\boldsymbol{Z}}(z)$ to be fitted is a normal *pdf* i.e. $\mathbb{f}_{\boldsymbol{Z}}(z) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{1}{2}\left(\frac{z_i - m}{\sigma}\right)^2\right)$ with the parameters m and σ^2 unknown. One method of solving for the parameters is by maximum likelihood estimation (MLE). In this method, one forms a maximum log-likelihood function given by:

$$l(\boldsymbol{\theta}; z) = \log \sum_{i=1}^{N} \mathbb{f}_{\boldsymbol{Z}}(z_i; \boldsymbol{\theta}) = -\frac{N}{2} log(2\pi\sigma^2) - \frac{1}{2}\sum_{i=1}^{N} \frac{(z_i - m)^2}{\sigma^2} \qquad \text{(E1.6)}$$

Here, $\boldsymbol{\theta} = \left(m, \sigma^2\right)^T$. The objective of the MLE is to find an estimate for $\boldsymbol{\theta}$ that maximizes $l(\boldsymbol{\theta}; \boldsymbol{z})$ with respect to $\boldsymbol{\theta}$. The estimate is meant to ensure that the observed data $\boldsymbol{z}$ is most likely to have been realized from the assumed *pdf*. Here, solve for the optimum estimate of the parameters $\boldsymbol{\theta}$ and check the sufficient conditions for optimality.

[Hint: Minimize the negative log-likelihood function $-l(\boldsymbol{\theta}; z)$ and get the estimate by solving the optimality conditions: $\frac{\partial l}{\partial m} = 0$ and $\frac{\partial l}{\partial \sigma^2} = 0$ yielding the familiar optimum estimates: $m = \frac{1}{N}\sum_{i=1}^{N}\left(z_i - m\right)$ which is the mean value of the observations and $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\left(z_i - m\right)^2$ which is the variance.]

8. In matrix eigenvalue analysis, minimization of Rayleigh quotient (Clough and Penzien 1982) yields the lowest eigenvector and the associated eigenvalue. It is a constrained optimization problem:

$$\text{min imize } \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x}$$

$$\text{s.t. } \boldsymbol{x}^T \boldsymbol{x} = 1 \qquad \text{(E1.7a,b)}$$

Assume that $\boldsymbol{A}$ is a 2x2 matrix, given by $\begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix}$. The constraint implies orthogonality of the eigenvectors. Solve the problem by Lagrange multiplier method. [Solution: $\boldsymbol{x}^* = \left(0.8112, -0.5847\right)^T$ and the corresponding eigenvalue = -0.1623]

NOTES

1 Readers may be familiar with basic probability theory (Ang and Tang 1984, Papoulis 1991, Roy and Rao 2017). However, some elements of the probability theory are provided in Appendix 1. See Probabilistic Route, p. 16.

2 An evolute of a curve is the locus of the centres of its curvatures. The curvature κ is defined as: $\kappa = \frac{d\phi}{ds}$ where ϕ is the tangent angle (with $tan\phi = \frac{dy}{dx}$) and s is the arc length (with $ds = \sqrt{dx^2 + dy^2}$). See Evolute, p. 27.

3 A conservative system is one wherein the set of forces inducing motion are conservative. We now give the definition of a conservative force.
Consider the work done by a force $\boldsymbol{F}$ in moving along a path in a domain Ω : it is given by the path integral of the so called differential work đW, i.e.:

$$\text{Work done} = \int_c đW = \int_c \boldsymbol{F}.d\boldsymbol{x}$$

Suppose that $\boldsymbol{F} = -\nabla W$; then the differential đW = dW is exact. Here the work done is independent of the path and depends only on the end points. Thus, a conservative force is derivable from the gradient of a work potential function W such that $F = \nabla W = \left(\frac{\partial W}{\partial x}, \frac{\partial W}{\partial y}, \frac{\partial W}{\partial z}\right)^T$. Since curl of a gradient is zero, for a conservative force $\boldsymbol{F}$ it is also true that $\nabla \times \boldsymbol{F} = 0$. Also, if curve c is closed, the path integral of the exact work differential reduces to zero irrespective of the path. If đW is not exact, it is referred to as an inexact differential. See Conservative system, p. 31.

4 Inner product of two functions u and v is given by $\langle u, v \rangle = \int_{\mho} u(\boldsymbol{x}) v(\boldsymbol{x}) d\boldsymbol{x}$. See Inner product, p. 35.

5 norm $\|u\|$ of a scalar-valued function $u(\boldsymbol{x})$ over a domain $\mho$:

$\|u\| = \sqrt{(u,u)} = \sqrt{\int_{\mho} u^2 d\boldsymbol{x}}, \forall u \in L^2(\mho)$ where $L^2(\mho)$ is the familiar notation for the set of all square integrable functions $v(\boldsymbol{x})$, i.e $\int_{\mho} v(\boldsymbol{x})^2 \, d\boldsymbol{x} < \infty$. See Norm, p. 35.

6 A basis B for a polynomial vector space $P = \{p_1, p_2, \ldots, p_n\}$ is a set of polynomials that spans the space and is linearly independent. p_n is a polynomial of degree n. $span(B) = P$ means that if $B = \{v_1, v_2, \ldots, v_n\}$, then every vector p in P can be uniquely expressed in the form:

$$p = \sum_{j=1}^{n} \alpha_j v_j, \; \alpha_j \in \mathbb{R}$$

The simplest possible basis for P is the monomial basis $\{1, x, x^2, x^3, \ldots, x^n\}$. See Polynomial basis set, p. 37.

REFERENCES

Ang, A. H. S. and W. H. Tang. 1984. *Probability Concepts in Engineering Planning and Design, Volume II Decision, Risk and Reliabilfity (Vol. II)*. John Wiley and Sons, Inc. NY.

Basdevant, J. 2007. *Variational Principles in Physics*, Springer, NY, USA.

Bathe, K. J. 1996. *Finite Element procedures*. Prentice-Hall International, Inc., Englewood Cliffs, NJ, USA.

Beichl, I. and F. Sulliman. 2000. The Metropolis algorithm, *Journal of Computing in Science and Engineering* 2(1): 65–69.

Belegundu, A. D. and T. R. Chadrupatla. 1999. *Optimization Concepts and Applications in Engineering*. Prentice Hall.

Bernoulli, J. 1697. Bending of light rays in transparent non-uniform media and the solution to the problem of determining the Brachistochrone curve. *Acta Eruditorum* 19: 206–211.

Binder, K. Applications of Monte Carlo methods to statistical physics. *Reports in Progress in Physics* 60: 487–559.

Bodin, L., B. L., A. A. Golden, M. Assad, and M. Ball. 1983. Routing and scheduling of vehicles and crews. The State of the Art Computers *and Operations Research* 10: 63–211.

Bonomi, E. and J. L. Lutton. 1984. The N-city travelling salesman problem: statistical mechanics and the Metropolis algorithm. *SIAM Review* 26: 551–568.

Cassel, K. W. 2013. *Variational Methods with Applications in Science and Engineering*. Cambridge University Press. NY. USA.

Chorin, A. J. and J. E. Marsden. 1993. *A Mathematical Introduction to Fluid Mechanics*. 3rd Ed. Springer Verlag. NY.

Clough, R. W. and J. Penzien. 1982. *Dynamics of Structures*. McGraw-Hill.

Cordeau, J. F., G. Esaulniers, J. Desrosiers, M. Solomon, and F. Soumis. 2002. The vehicle routing problem with time windows. In: *The Vehicle Routing Problem*. Toth, P. & Vigo, D. (eds.), pp. 157–193. SIAM Publishing. Philadelphia.

Cornuejols, G. and R. Tütüncü. 2007. *Optimization Methods in Finance*. Cambridge University Press. UK.

Courant, R. 1943. Variational method for the solution of problems equilibrium and vibration. *Bulletin of the AMS* 49: 1–23.

Curiel, I. 1997. *Cooperative Game Theory and Applications, Cooperative Games Arising from Combinatorial Optimization Problems.* Kluwer Academic Publishers. The Netherlands.

Dantzig, G. B., R. Fulkerson, and J. Johmson. 1954. Solution of a large-scale traveling salesman problem. *Journal of the Operations Research Society of America* 2: 393–410.

Dantzig, G. B. and J. H. Ramser. 1959. The truck dispatching problem. *Management Science* 6(1): 80–91.

dell'Isola, F. and S. Gavrilyuk. (eds.). 2011. *Variational Models and Methods in Solid and Fluid Mechanics, CISM courses and lectures*. Springer.

Derigs, U. (Editor). 2009. *Optimization and Operations Research*. EOLSS Publishers Co Ltd. Oxford, UK.

Diaconis, P. and L. Saloff-Coste. 1998. What do we know about the Metropolis algorithm? Journal *of Computer and System Sciences* 57: 20–36.

Dongarra, J. and F. Sullivan. 2000. Guest editors introduction to the top 10 algorithms. *Computing in Science and Engineering* 2: 22–23.

Dulebenets, M. A. 2018. A comprehensive evaluation of weak and strong mutation mechanisms in evolutionary algorithms for truck scheduling at cross-docking terminals. *IEEE Access* 6: 65635–65650.

Dulebenets, M. A. 2019. A delayed start parallel evolutionary algorithm for just-in-time truck scheduling at a cross-docking facility. *International Journal of Production Economics* 212: 236–258.

Dym, C. L. and I. H. Shames. 2013. *Solid Mechanics, A Variational Approach*. Springer. NY, USA.

Erlichson, H. 1999. Johann Bernoulli's brachistochrone solution using Fermat's principle of least time. *European Journal of Physics* 20.

Finlayson, B. A., 1972. *The Method of Weighted Residuals and Variational Principles.* Academic Press, NY.

Fletcher, R. and C. M. Reeves. 1964. Function minimization by conjugate gradients. *Computer Journal* 7: 149–154.

Fogel, D. B., 1988. An evolutionary approach to the travelling salesman problem, Biological Cybernetics 60: 139–144.

Garfinkel, R. S., 1985. Motivation and modeling. In: Lawer, E. L., Lenstra, J. K., RinnooyKan, A. H. G. and Shmoys, D. B. (eds.) *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, pp. 17–36. John Wiley and Sons. NY.

Garg, R. 2008. *Analytical and Computational Methods in Electromagnetics*. Artech House, Inc. MA.

Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st Ed. Addison-Wesley Publishing Company. Boston.

Goldstine, H. H. 1980. *A History of the Calculus of Variations from the 17th Through the 19th Century*. Berlin. Springer.

Gould, H. and J. Tobochnik. 2010. *Statistical and Thermal Physics: With Computer Applications*. Princeton University Press. NJ.

Govindan, K., A. Jafarian, and V. Nourbakhsh. 2018. Designing a sustainable supply chain network integrated with vehicle routing: A comparison of hybrid swarm intelligence metaheuristics. *Computers & Operations Research* 110:220–235.

Gutin, G. and A. P. Punnen. (eds.). 2002. The traveling salesman problem applications, formulations and variations. In: *The Traveling Salesman Problem and Its Variations*. Kluwer Academic Publishers. Dordrecht, The Netherlands.

Hrennikoff, A. 1941. Solution of problems of elasticity by the framework method. *Journal of Applied Mechanics* 8(4): 169–175.

Hughes, T. J. R. 2012. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. NY: Dover Publications.

Jackson, J.D. 1999. *Classical Electrodynamics,* 3rd ed. Wiley. NY.

Janke, W. 2008. Monte Carlo methods in classical statistical physics. Lecture *Notes in Physics* 739: 79–140.

Karush, W. 1939. *Minima of Functions of Several Variables with Inequalities as Side Conditions, Master's Thesis*. Department of Mathematics. University of Chicago.

Kirk, D. E. 1970. *Optimal Control Theory, An Introduction*. Dover Publication, Inc., Mineola, New York.

Khun, H. W. and A. W. Tucker. 1951. Nonlinear programming. In: J. Neyman (ed.) *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492. University of California Press, Berkeley.

Landau, D. P. and K. Binder. 2000. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press. NY.

Laporte, G., 1992a. The vehicle routing problem: an overview of exact and approximative algorithms. *European Journal of Operational Research* 59(3): 345–358.

Laporte, G. 1992b. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* 59: 231–247.

Lawler, E. L., J. K. Lenstra, A. H. G. Rinnooy-Kan, and D. B. Shmoys. 1985. *The Traveling Salesman Problem*. John Wiley and Sons. NY.

Lenstra, J. K. and A. H. G. RinnooyKan. 1975. Some simple applications of the travelling salesman problem. Operational *Research Quarterly* 26(4): 717–733.

Liberzon, D., 2012. *Calculus of Variations and Optimal Control Theory, A concise Introduction*. Princeton University Press. NJ.

Meirovitch, L., 1967. *Methods of Analytical Dynamics*, Macmillan, NY.

Meirovitch, L. 1970. *Methods of Analytical Dynamics*. McGraw-Hill, NY.
Meirovitch, L. 1990. *Dynamics and Control of Structures*. John Wiley & Sons, Inc.
Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. 1953. Equation of state calculations by fast computing machines. *Journal of Chemical Physics* 21(6): 1087–1092.
Nicholson, T. A. J. 1971. *Optimization in Industry: Industrial Applications*. London Business School Series. Transaction Publishers. New Brunswick, USA.
Nocedal, J. and S. J. Wright. 2006. *Numerical Optimization*. Springer-Verlag, NY.
O'Connell, J. P. and J. M. Haile. 2005. *Thermodynamics: Fundamentals for Applications*. Cambridge University Press. NY.
Papadimitriou, C. 1994. *Computational Complexity*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA. USA.
Papadimitriou, C. and K. Steiglitz. 1982. *Combinatorial Optimization, Algorithms and Complexity*. Prentice Hall. NJ.
Papadimitriou, C. and M. Yannakakis. 1991. Optimization, approximation, and complexity classes. Journal of Computer and System Sciences 43(3): 425–440.
Papoulis, A., 1991. *Probability, Random Variables and Stochastic Processes*. 3rd Ed. McGraw-Hill, Inc. NY.
Pontryagin, L. S., V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. 1962. The Mathematical Theory of Optimal Processes, English translation, Interscience.
Rardin, R. L. 1997. *Optimization in Operations Research*. 1st Ed. Pearson.
Reddy, J. N. 2002. *Energy Principles and Variational Methods in Applied Mechanics*. 2nd Ed. John Wiley and Sons, Inc. NJ, USA.
Reinelt, G., 1994. *The Traveling Salesman: Computational Solutions for TSP Applications*. Springer-Verlag. Berlin, Heidelberg.
Roy, D. and Rao, G. V. 2012. *Elements of Structural Dynamics: A New Perspective*. John Wiley & Sons.
Roy, D. and G.V. Rao. 2017. *Stochastic Dynamics, Filtering and Optimization*. Cambridge University Press. UK.
Rudin, W. 1976. *Principles of Mathematical Analysis*. 3rd ed. NY: McGraw-Hill.
Stengel, R.F. 1994. *Optimal Control and Estimation*. 2nd ed. NY: Dover Publication Inc.
Strang, G. and G. Fix. 1973. *An Analysis of The Finite Element Method*. Prentice Hall.
Struik D. J (ed). 1986. *A Source Book in Mathematics*. Princeton. Princeton University Press. NJ.
Tanizaki, H. 2004. *Computational Methods in Statistics and Econometrics*. Marcel Dekker, Inc. NY.
TerHaar, D., 1971. *Elements of Hamiltonian Mechanics*, Pergamon Press, Oxford.
Troutman, J. L. 1996. *Variational Calculus and Optimal Control: Optimization with Elementary Convexity*. 2nd Ed. Springer. NY.
Vahdani, B. and S. Shahramfard. 2019. A truck scheduling problem at a cross-docking facility with mixed service mode dock doors. Engineering Computations 12: 648.
von Neumann, J. 1928. On the theory of games of strategy. A translation by Mrs. Sonya Bargmannof "zur Theorie der Gesellschaftsspiele". *Mathematische Annalen* 100:. 295–320.
von Neumann, J. and Morgenstern, O. 1953. *Theory of Games and Economic Behavior*. Princeton University Press.
Zienkiewicz, O. C., 1977. *The Finite Element Method*, 3rd Ed. McGraw-Hill Book Co., London.

BIBLIOGRAPHY

Applegate, D. L., Bixby, R. E., Chavtal, V and W. J. Cook. 2007. *The Traveling Salesman Problem: A Computational Study*. (Princeton Series in Applied Mathematics). 2nd Ed. Princeton University Press.

Cook, W. J., W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. 1997. *Combinatorial Optimization*. John Wiley and Sons, Inc. NY.

Dennis, J. E. and R. B. Schnabel. 1993. *Numerical Methods for Unconstrained Optimization*. (Prentice Hall. Englewood Cliffs. NJ. 1983). Reprinted by SIAM Publications.

Fletcher, R.1987. *Practical Methods of Optimization*.2nd Ed., Wiley. NY.

Parlaktun, O., A. Sipahioˇglu, A. Yazıcı, and U. G¨urel. 2005. Tsp Approach For Mobile Robot Dynamic Path Planning. *The 1st International Conference on Control and Optimization with Industrial Application*. Abstracts. Baku (Azerbaijan).

Patrick, B. 2012. *Optimization for industrial problems*. Springer.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. 2007. *Numerical Recipes: The Art of Scientific Computing*. 3rd Ed. Cambridge University Press. NY.

Snyman, J. A. 2005. *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. Springer.

S, Arora. 1998. The Approximability of NP-hard Problems. Princeton University Press.

2 Classical Derivative-based Optimization Techniques

2.1 INTRODUCTION

If we were to track how optimization methods developed and matured, we would typically come across a rag tag collection of techniques that evolved over the last century, starting with the simple and less robust ad hoc methods, e.g. the brute force search methods, to the more rigorously founded derivative-based and direct search methods. Brute force methods (Section 1.3.1, Chapter 1) apply a trial-and-test scheme to all possible candidates at each iteration. The obvious disadvantage is that the search space may be prohibitively large even for an apparently simple problem as illustrated in Chapter 1 – see our discussion on the travelling salesman problem. They offer freedom from computing derivatives and are thus easy to implement, perhaps at the cost of slower convergence as illustrated in Chapter 1. However, with the availability of faster computing machines that have encouraged development of algorithms to numerically evaluate functional derivatives, derivative-based (gradient search) methods have gained prominence in all fields – science, engineering and finance. The steepest descent (Cauchy 1847, Curry 1944), conjugate gradient (Hestenes and Stiefel 1952, Fletcher and Reeves 1964, Fletcher 1976), Newton and quasi-Newton methods (Davidon 1959, Broyden 1967) belong to this category. These methods are primarily meant for solving unconstrained optimization problems. Note that most constrained optimization problems are conveniently solved by transforming them to unconstrained ones. This is illustrated in Section 1.6, Chapter 1, whilst describing the method of Lagrange multipliers. The optimality conditions for both unconstrained and constrained optimization problems are also described in Chapter 1. Development of the derivative-based methods and their variants, such as the BFGS method (Broyden 1970, Fletcher 1970, Goldfarb 1970, Shanno 1970) and DFP (Davidon 1959, Fletcher and Powell 1963), is mostly guided by the requirement to improve the search direction using the function derivatives so as to achieve a better convergence rate. The present chapter focusses on describing a few such optimization techniques – mainly derivative-based.

2.2 BASIC GRADIENT METHODS

It has been shown in Chapter 1 (Section 1.6) that if $f(\boldsymbol{x}): \mathbb{R}^n \rightarrow \mathbb{R}$ is the scalar objective function, $-\nabla f(\boldsymbol{x})$ is the direction of steepest descent and $\nabla f(\boldsymbol{x})$ that of

DOI: 10.1201/9781003414063-2

steepest ascent. Gradient based methods utilize this information to iteratively reach the optimum point $\boldsymbol{x}^*$.

2.2.1 Steepest Descent Method (Cauchy 1847)

With $\boldsymbol{x}_0$ as the starting point, the steepest descent method uses the following updating procedure at the k^{th} iteration:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - s_k \nabla f\left(\boldsymbol{x}_k\right) \tag{2.1}$$

$s_k \in \mathbb{R}^+$ is the step size in the direction $\boldsymbol{d}_k = -\nabla f\left(\boldsymbol{x}_k\right)$. The step size is conveniently found by minimizing $f\left(\boldsymbol{x}\right)$ in the direction $-\nabla f\left(\boldsymbol{x}_k\right)$, i.e.:

$$s_k := \arg\min_{s_k \in \mathbb{R}} \left(f(\boldsymbol{x} - s_k \nabla f\left(\boldsymbol{x}_k\right)\right) = \arg\min_{s_k} h\left(s_k\right) \tag{2.2}$$

With $h\left(s_k\right) = f\left(\boldsymbol{x} - s_k \nabla f\left(\boldsymbol{x}_k\right)\right)$, Equation (2.2) constitutes a line search – a one dimensional optimization of $h\left(s_k\right)$. Any interval bracketing technique such as golden section, Fibonacci search (see Exercise 2 in Chapter 1) or an interpolation technique like quadratic or cubic fit (Belegundu and Chandrupatla 1999) may be employed for the purpose. The steepest descent method, though not computationally efficient, offers guaranteed convergence for a class of functions (Bertsekas 1996, Nocedal and Wright 2006).

For a function $f\left(\boldsymbol{x}\right)$, if the gradient $\nabla f\left(\boldsymbol{x}\right)$ is M-Lipschitz continuous* i.e.:

$$\left\| \nabla f\left(\boldsymbol{y}\right) - \nabla f\left(\boldsymbol{x}\right) \right\| \le M \left\| \left(\boldsymbol{y} - \boldsymbol{x}\right) \right\|, M \ge 0 \tag{2.3}$$

then for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$,

$$f\left(\boldsymbol{y}\right) \le f\left(\boldsymbol{x}\right) + \nabla f\left(\boldsymbol{x}\right)^T \left(\boldsymbol{y} - \boldsymbol{x}\right) + \frac{M}{2} \left\| \left(\boldsymbol{y} - \boldsymbol{x}\right) \right\|^2 \tag{2.4}$$

---

* *M-Lipschitz continuous function*
Consider a function $g\left(\boldsymbol{x}\right): \mathbb{R}^n \to \mathbb{R}$. It is Lipschitz continuous with Lipschitz constant $M \ge 0$ if for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$

$$\left\| g\left(\boldsymbol{y}\right) - \mathrm{g}\left(\boldsymbol{x}\right) \right\| \le M \left\| \left(\boldsymbol{y} - \boldsymbol{x}\right) \right\| \tag{i}$$

By first-order Taylor approximation, $g\left(\boldsymbol{y}\right) \cong g\left(\boldsymbol{x}\right) + \nabla g\left(\boldsymbol{z}\right)^T \left(\boldsymbol{y} - \boldsymbol{x}\right)$, for a $\boldsymbol{z}$ lying on the straight line with $\boldsymbol{x}$ and $\boldsymbol{y}$ as boundary points. If $\left\| \nabla g\left(\boldsymbol{z}\right) \right\| \le M$ ($M \ge 0$), one gets the inequality in Equation (i).

Proof of the inequality in the last equation is provided in the footnote below.†

$\|.\|$ denotes the Euclidean or L^2 norm, i.e. $\|(\boldsymbol{y}-\boldsymbol{x})\| = \sqrt{(\boldsymbol{y}-\boldsymbol{x})^T(\boldsymbol{y}-\boldsymbol{x})}$. The quadratic upper bound in Equation (2.4) amounts to a bound on the value of $f(\boldsymbol{x})$ at the end of the k^{th} iteration. Thus:

$$f(\boldsymbol{x}_{k+1}) \le f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^T(\boldsymbol{x}_{k+1}-\boldsymbol{x}_k) + \frac{M}{2}\|(\boldsymbol{x}_{k+1}-\boldsymbol{x}_k)\|^2$$

$$\Rightarrow f(\boldsymbol{x}_{k+1}) \le f(\boldsymbol{x}_k) - s_k \nabla f(\boldsymbol{x}_k)^T \nabla f(\boldsymbol{x}_k) + \frac{Ms_k^2}{2}\|\nabla f(\boldsymbol{x}_k)\|^2$$

$$\Rightarrow f(\boldsymbol{x}_{k+1}) \le f(\boldsymbol{x}_k) - s_k\left(1-\frac{Ms_k}{2}\right)\|\nabla f(\boldsymbol{x}_k)\|^2 \qquad (2.5a)$$

With a step size small enough, say $s_k < \frac{1}{M}$, one has:

$$\left(1-\frac{Ms_k}{2}\right) \ge \frac{1}{2} \text{ and } f(\boldsymbol{x}_{k+1}) \le f(\boldsymbol{x}_k) - \frac{s_k}{2}\|\nabla f(\boldsymbol{x}_k)\|^2 \qquad (2.5b)$$

Since $\frac{s_k}{2}\|\nabla f(\boldsymbol{x}_k)\|^2$ is always positive (unless $\nabla f(\boldsymbol{x}_k) = 0$), it implies that $f(\boldsymbol{x})$ decreases with iterations until the optimum is reached, thus ensuring the convergence of the descent method when the step size is small.

Even if f is non-quadratic, the iterative process shows local progress, provided s_k is sufficiently small. It is helpful to know that the convergence rate of the gradient method is related to the condition number of the Hessian matrix $\boldsymbol{H} = \nabla^2 f(\boldsymbol{x})$ which is symmetric (Bertsekas 1996). The explanation below clarifies the point.

---

† Proof of the inequality in Equation (2.4)
Since, by assumption, $\nabla f(\boldsymbol{x})$ is M-Lipschitz continuous, $\nabla^2 f(\boldsymbol{x}) \le M\mathbf{I}$ and for $\forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$, one has:

$$(\boldsymbol{y}-\boldsymbol{x})^T \nabla^2 f(\boldsymbol{z})(\boldsymbol{y}-\boldsymbol{x}) \le M\|(\boldsymbol{y}-\boldsymbol{x})\|^2 \qquad \text{(i)}$$

Now, $\forall \boldsymbol{x}, \boldsymbol{y}$ and $\boldsymbol{z} \in [\boldsymbol{x}, \boldsymbol{y}]$, we have the quadratic approximation (by Taylor's expansion):

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y}-\boldsymbol{x}) + \frac{1}{2}(\boldsymbol{y}-\boldsymbol{x})^T \nabla^2 f(\boldsymbol{z})(\boldsymbol{y}-\boldsymbol{x})$$

$$\le f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y}-\boldsymbol{x}) + \frac{1}{2}M\|(\boldsymbol{y}-\boldsymbol{x})\|^2 \qquad \text{(ii)}$$

which proves the inequality.

Refer to the definition of a convex function in Chapter 1 (Section 1.2). For a strictly convex function, $\boldsymbol{H}$ is positive definite with all its eigenvalues real and positive. Condition number of $\boldsymbol{H}$ is the ratio of the highest to the lowest eigenvalue. A strongly convex function is also characterized by a lower bound (Boyd and Vandenberghe 2004), i.e.:

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T (\boldsymbol{y}-\boldsymbol{x}) + \frac{m}{2}\|(\boldsymbol{y}-\boldsymbol{x})\|^2, m \geq 0 \tag{2.6}$$

From these upper and lower bounds in Equations (2.5) and (2.6) respectively, it follows that:

$$\frac{m}{2}\|(\boldsymbol{y}-\boldsymbol{x})\|^2 \leq f(\boldsymbol{y}) - \left(f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T (\boldsymbol{y}-\boldsymbol{x})\right) \leq \frac{M}{2}\|(\boldsymbol{y}-\boldsymbol{x})\|^2 \tag{2.7}$$

(2.7) indicates that if $f(\boldsymbol{x})$ is twice differentiable,

$$\frac{m}{2}\|(\boldsymbol{y}-\boldsymbol{x})\|^2 \leq (\boldsymbol{y}-\boldsymbol{x})^T \boldsymbol{H}(z)(\boldsymbol{y}-\boldsymbol{x}) \leq \frac{M}{2}\|(\boldsymbol{y}-\boldsymbol{x})\|^2 \tag{2.8}$$

for some $\boldsymbol{z}$ on the line between $\boldsymbol{x}$ and $\boldsymbol{y}$. From the definition of the eigenvalue problem for $\boldsymbol{H}$ in the form $\boldsymbol{H}\boldsymbol{\Phi} = \lambda\boldsymbol{\Phi}$ where λ is an eigenvalue and $\boldsymbol{\Phi}$ the corresponding eigenvector, (2.8) shows that all the eigenvalues of $\boldsymbol{H}$ lie between m and M, i.e:

$$\Rightarrow m \leq \frac{\boldsymbol{\Phi}^T \boldsymbol{H}(z)\boldsymbol{\Phi}}{\|\boldsymbol{\Phi}\|^2} \leq M \tag{2.9}$$

Here, $\frac{M}{m}$ stands for the condition number of the matrix $\boldsymbol{H}$. If the condition number is close to one, the matrix is well-behaved and $f(\boldsymbol{x})$ is of low convexity. Otherwise, $\boldsymbol{H}$ is ill-conditioned and $f(\boldsymbol{x})$ is of strong convexity with the effect that the gradient method may have low convergence rate. For the quadratic function in Figure 2.1a, we observe that the condition number is unity (the two eigenvalues of $\boldsymbol{H}$ are the same) and the optimum is reached in a single step. The condition number is three for the function in Figure 2.1b and the gradient method took ten iterations to reach the optimum.

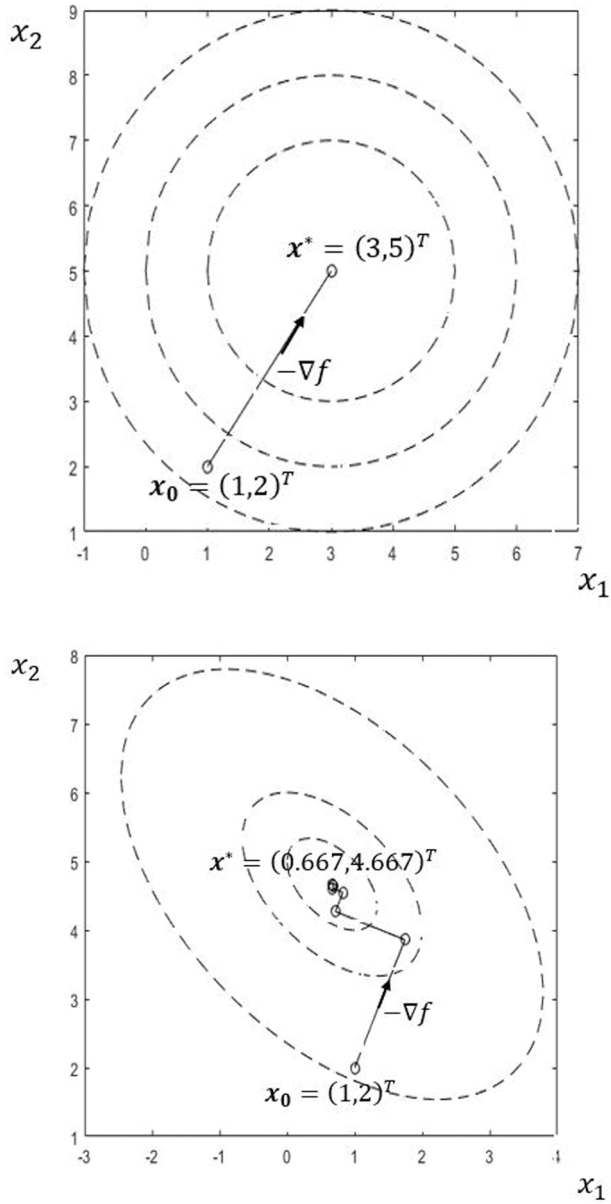


FIGURE 2.1 Convergence of steepest descent method for quadratic functions; $\boldsymbol{x}_0$ – starting point, $\boldsymbol{x}^*$ – optimum point: (a) $f(x_1, x_2) = (x_1 - 3)^2 + (x_2 - 5)^2$ – optimum realized in one iteration and (b) $f(x_1, x_2) = (x_1 - 3)^2 + (x_2 - 5)^2 + x_1 x_2$ – optimum realized in ten iterations.

2.2.2 Conjugate Gradient Method

The conjugate gradient (CG) method (Hestenes and Stiefel 1952,‡ Fletcher and Powell 1963) achieves convergence of an n-dimensional quadratic function exactly in n steps (Nocedal and Wright 2006). To show this, consider the function as:

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + c \tag{2.10}$$

$\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ is a symmetric matrix. $\boldsymbol{b} \in \mathbb{R}^n$ and $c \in \mathbb{R}$. The gradient to $f(\boldsymbol{x})$ is:

$$\nabla f(\boldsymbol{x}) = \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{b} \tag{2.11}$$

Denote this gradient vector at the k^{th} iteration by $\boldsymbol{g}_k = \boldsymbol{Q}\boldsymbol{x}_k + \boldsymbol{b}$. CG method envisages availability of n directions $\boldsymbol{d}_i, i = 0,1,\ldots,n-1$ which are $\boldsymbol{Q}$-conjugate, i.e:

$$\boldsymbol{d}_i^T\boldsymbol{Q}\boldsymbol{d}_j = 0, i \neq j \tag{2.12}$$

With $\boldsymbol{x}_0$ as the starting vector and $\boldsymbol{x}^*$ the optimum with $\nabla f(\boldsymbol{x}^*) = \boldsymbol{Q}\boldsymbol{x}^* + \boldsymbol{b} = 0$, one can express the vector $\boldsymbol{x}^* - \boldsymbol{x}_0$ as a linear combination of the conjugate directions:

$$\boldsymbol{x}^* - \boldsymbol{x}_0 = \sum_{i=0}^{n-1}\alpha_i\boldsymbol{d}_i \tag{2.13}$$

Pre-multiplying both sides of the last equation by $\boldsymbol{d}_i^T\boldsymbol{Q}$, and utilizing the conjugacy property in Equation (2.12), we obtain the coefficients α_i as:

$$\alpha_i = \frac{\boldsymbol{d}_i^T\boldsymbol{Q}(\boldsymbol{x}^* - \boldsymbol{x}_0)}{\boldsymbol{d}_i^T\boldsymbol{Q}\boldsymbol{d}_i} \tag{2.14}$$

Similarly, during the iteration process, if $\boldsymbol{x}_k$ is update at $(k-1)^{th}$ iteration with known step sizes $s_i, i = 1,2,\ldots,k-1$, we also have:

$$\boldsymbol{x}_k - \boldsymbol{x}_0 = \sum_{i=0}^{k-1}s_i\boldsymbol{d}_i$$

$$\Rightarrow \boldsymbol{d}_k^T\boldsymbol{Q}(\boldsymbol{x}_k - \boldsymbol{x}_0) = 0 \tag{2.15}$$

---

‡ *Hestenes and Stiefel [1952]*

If we consider $\nabla f(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$, optimization of the quadratic function $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x} + c$ is equivalent to solving $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. Hestenes and Stiefel in their seminal paper [1952] first introduced the iterative CG as an effective method superior to Gaussian elimination for solving a system of n simultaneous equations when n is large.

Therefore,

$$\boldsymbol{d}_k^T \boldsymbol{Q}\left(\boldsymbol{x}^* - \boldsymbol{x}_0\right) = \boldsymbol{d}_k^T \boldsymbol{Q}\left(\boldsymbol{x}^* - \boldsymbol{x}_k + \boldsymbol{x}_k - \boldsymbol{x}_0\right)$$

$$= \boldsymbol{d}_k^T \boldsymbol{Q}\left(\boldsymbol{x}^* - \boldsymbol{x}_k\right)$$

$$= \boldsymbol{d}_k^T \left(-\boldsymbol{b} - \boldsymbol{Q}\boldsymbol{x}_k\right) \quad (\text{since } \boldsymbol{Q}\boldsymbol{x}^* + \boldsymbol{b} = \boldsymbol{0})$$

$$= -\boldsymbol{d}_k^T \nabla f\left(\boldsymbol{x}_k\right) \quad \left(\text{since } \nabla f\left(\boldsymbol{x}_k\right) = \boldsymbol{Q}\boldsymbol{x}_k + \boldsymbol{b}\right)$$

$$\Rightarrow \alpha_k = -\frac{\boldsymbol{d}_k^T \nabla f\left(\boldsymbol{x}_k\right)}{\boldsymbol{d}_k^T \boldsymbol{Q}\boldsymbol{d}_k} \ \left(\text{from Equation 2.14}\right) \tag{2.16}$$

The RHS in the last step of Equation (2.16) is in fact the optimum step size s_k to get the update $\boldsymbol{x}_{k+1}$. This is since (i) $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + s_k \boldsymbol{d}_k$ and (ii) one finds s_k by line search by minimizing $f\left(\boldsymbol{x}\right)$ in the direction $\boldsymbol{d}_k$ (see Equations 2.1 and 2.2). This proves the assertion that CG converges in exactly n steps for a quadratic function. It now remains to find the conjugate directions $\boldsymbol{d}_k, k = 0,1,\ldots$ along with the step sizes $s_k, k = 0,1,\ldots$ as iterations progress.

Zeroth iteration

Suppose that we choose, at the zeroth iteration, an arbitrary direction $\boldsymbol{d}_0$ and proceed with a step size s_0 to obtain:

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + s_0 \boldsymbol{d}_0 \tag{2.17}$$

s_0 is obtained from Equation (2.16) as:

$$\Rightarrow s_0 = -\frac{\boldsymbol{d}_0^T \boldsymbol{g}_0}{\boldsymbol{d}_0^T \boldsymbol{Q}\boldsymbol{d}_0} \left(\text{since } \nabla f\left(\boldsymbol{x}_0\right) = \boldsymbol{g}_0\right) \tag{2.18a}$$

At the start of the iterations, $\boldsymbol{d}_0$ is often taken as $\boldsymbol{g}_0$ and hence s_0 is:

$$s_0 = -\frac{\boldsymbol{g}_0^T \boldsymbol{g}_0}{\boldsymbol{g}_0^T \boldsymbol{Q}\boldsymbol{g}_0} \tag{2.18b}$$

Iteration k=1

$$\boldsymbol{g}_1 = \nabla f\left(\boldsymbol{x}_1\right) = \boldsymbol{Q}\boldsymbol{x}_1 + \boldsymbol{b}$$

$$= \boldsymbol{Q}\left(\boldsymbol{x}_0 + s_0 \boldsymbol{d}_0\right) + \boldsymbol{b} = \boldsymbol{g}_0 + s_0 \boldsymbol{Q}\boldsymbol{d}_0 \tag{2.19}$$

Now, define:

$$\boldsymbol{x}_2 = \boldsymbol{x}_1 + s_1 \boldsymbol{d}_1 \tag{2.20}$$

where s_1 is obtained as follows:

$$s_1 = -\frac{\boldsymbol{d}_1^T \boldsymbol{g}_1}{\boldsymbol{d}_1^T \boldsymbol{Q} \boldsymbol{d}_1} = -\frac{\boldsymbol{g}_1^T \boldsymbol{d}_1}{\boldsymbol{d}_1^T \boldsymbol{Q} \boldsymbol{d}_1} \quad \left(\text{since } \nabla f\left(\boldsymbol{x}_1\right) = \boldsymbol{g}_1\right) \tag{2.21}$$

The unknown $\boldsymbol{d}_1$ in the last equation is assumed as:

$$\boldsymbol{d}_1 = -\boldsymbol{g}_1 + \beta_0 \boldsymbol{d}_0 \tag{2.22}$$

β_0 is a scalar and is obtained by seeking $\boldsymbol{d}_1$ to be $\boldsymbol{Q}$-conjugate to $\boldsymbol{d}_0$, i.e:

$$\boldsymbol{d}_1^T \boldsymbol{Q} \boldsymbol{d}_0 = 0$$

$$\Rightarrow \left(-\boldsymbol{g}_1 + \beta_0 \boldsymbol{d}_0\right)^T \boldsymbol{Q} \boldsymbol{d}_0 = 0$$

$$\Rightarrow \beta_0 = \frac{\boldsymbol{g}_1^T \boldsymbol{Q} \boldsymbol{d}_0}{\boldsymbol{d}_0^T \boldsymbol{Q} \boldsymbol{d}_0} = \frac{\boldsymbol{d}_0^T \boldsymbol{Q} \boldsymbol{g}_1}{\boldsymbol{d}_0^T \boldsymbol{Q} \boldsymbol{d}_0} \tag{2.23}$$

One observation is that $\boldsymbol{g}_1^T \boldsymbol{g}_0 = 0$. This is true since:

$$\boldsymbol{g}_1^T \boldsymbol{g}_0 = \left(\boldsymbol{g}_0 + s_0 \boldsymbol{Q} \boldsymbol{d}_0\right)^T \boldsymbol{g}_0 \left(\text{from Equation 2.19}\right)$$

$$= \boldsymbol{g}_0^T \boldsymbol{g}_0 + s_0 \boldsymbol{d}_0^T \boldsymbol{Q} \boldsymbol{d}_0 = \boldsymbol{g}_0^T \boldsymbol{g}_0 + s_0 \boldsymbol{g}_0^T \boldsymbol{Q} \boldsymbol{g}_0 \left(\text{since } \boldsymbol{d}_0 = \boldsymbol{g}_0\right)$$

$$= \boldsymbol{g}_0^T \boldsymbol{g}_0 - \frac{\boldsymbol{g}_0^T \boldsymbol{g}_0}{\boldsymbol{g}_0^T \boldsymbol{Q} \boldsymbol{g}_0} \boldsymbol{g}_0^T \boldsymbol{Q} \boldsymbol{g}_0 = 0 \left(\text{substituting for } s_0 \text{ from Equation 2.18b}\right) \tag{2.24}$$

Another observation is that $\boldsymbol{g}_1^T \boldsymbol{d}_0 = 0$, since $\boldsymbol{d}_0 = \boldsymbol{g}_0$.

The same procedure may be repeated for the rest of the iterations (Figure 2.2). However, in order to make certain general observations on the CG method, it is necessary to go through the steps for $k = 2$ also.

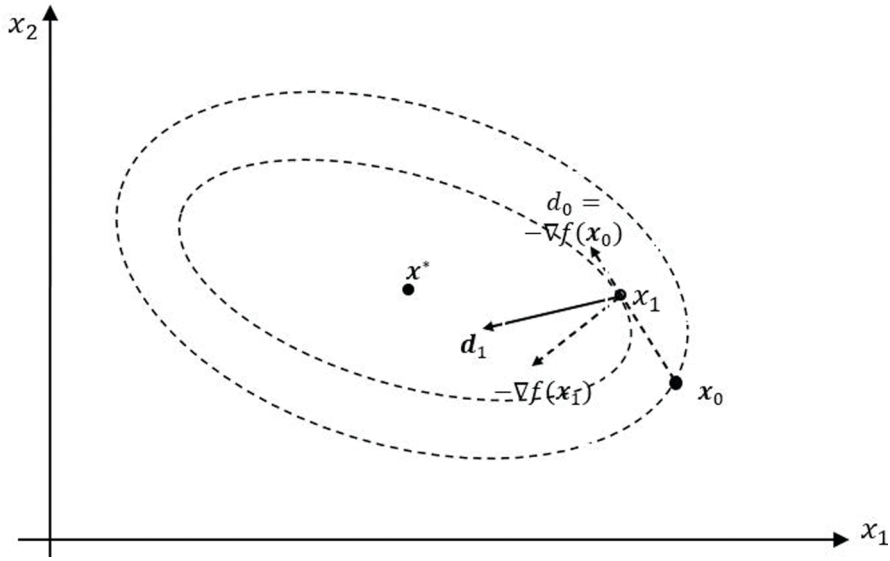


FIGURE 2.2 Conjugate gradient method, descent directions $-\nabla f(\boldsymbol{x}_0)$ and $-\nabla f(\boldsymbol{x}_1)$, conjugate directions $\boldsymbol{d}_0$ and $\boldsymbol{d}_1$ at zeroth and first iterations respectively.

Iteration k=2

The gradient and the update follow from Equations (2.19) and (2.20).

$$\boldsymbol{g}_2 = \nabla f(\boldsymbol{x}_2) = \boldsymbol{Q}\boldsymbol{x}_2 + \boldsymbol{b} = \boldsymbol{g}_1 + s_1\boldsymbol{Q}\boldsymbol{d}_1 \tag{2.25a}$$

$$\boldsymbol{x}_3 = \boldsymbol{x}_2 + s_2\boldsymbol{d}_2 \tag{2.25b}$$

The step size s_2 is given by:

$$s_2 = -\frac{\boldsymbol{d}_2^T\boldsymbol{g}_2}{\boldsymbol{d}_2^T\boldsymbol{Q}\boldsymbol{d}_2} = -\frac{\boldsymbol{g}_2^T\boldsymbol{d}_2}{\boldsymbol{d}_2^T\boldsymbol{Q}\boldsymbol{d}_2}\left(\text{see Equation 2.21}\right) \tag{2.25c}$$

As in the previous iteration, the new direction $\boldsymbol{d}_2$ is assumed to be:

$$\boldsymbol{d}_2 = -\boldsymbol{g}_2 + \beta_1\boldsymbol{d}_1 \tag{2.25d}$$

β_1 is a scalar constant and is obtained by seeking $\boldsymbol{d}_2$ to be $\boldsymbol{Q}$-conjugate to $\boldsymbol{d}_1$, i.e.:

$$\boldsymbol{d}_2^T\boldsymbol{Q}\boldsymbol{d}_1 = 0 \tag{2.25e}$$

which gives:

$$\beta_1 = \frac{\boldsymbol{g}_2^T \boldsymbol{Q}\boldsymbol{d}_1}{\boldsymbol{d}_1^T \boldsymbol{Q}\boldsymbol{d}_1} = \frac{\boldsymbol{d}_1^T \boldsymbol{Q}\boldsymbol{g}_2}{\boldsymbol{d}_1^T \boldsymbol{Q}\boldsymbol{d}_1} \quad \left(\text{see Equation 2.23}\right) \tag{2.25f}$$

This completes the 2^{nd} iteration. From the above two iterations, we may conclude that:

$$\boldsymbol{g}_2^T \boldsymbol{g}_1 = 0$$

ii) $\boldsymbol{g}_2^T \boldsymbol{d}_i = 0\,,\ i < 2$ (2.26a,b)

Proof of i):

$$\begin{aligned} \boldsymbol{g}_2^T \boldsymbol{g}_1 &= \left(\boldsymbol{g}_1 + s_1 \boldsymbol{Q}\boldsymbol{d}_1\right)^T \boldsymbol{g}_1 \left(\text{from Equation 2.25a}\right) \\ &= \boldsymbol{g}_1^T \boldsymbol{g}_1 + s_1 \boldsymbol{d}_1^T \boldsymbol{Q}\boldsymbol{g}_1 \end{aligned} \tag{2.27a}$$

Substituting for s_1 from Equation (2.21) and utilizing the relationship (2.22) between $\boldsymbol{g}_1$ and $\boldsymbol{d}_1$ gives:

$$\begin{aligned} \boldsymbol{g}_2^T \boldsymbol{g}_1 &= \boldsymbol{g}_1^T \boldsymbol{g}_1 - \frac{\boldsymbol{g}_1^T \left(-\boldsymbol{g}_1 + \beta_0 \boldsymbol{d}_0\right)}{\boldsymbol{d}_1^T \boldsymbol{Q}\boldsymbol{d}_1} \boldsymbol{d}_1^T \boldsymbol{Q}\left(-\boldsymbol{d}_1 + \beta_0 \boldsymbol{d}_0\right) \\ &= \boldsymbol{g}_1^T \boldsymbol{g}_1 + \boldsymbol{g}_1^T \left(-\boldsymbol{g}_1 + \beta_0 \boldsymbol{d}_0\right) \\ &= 0 \left(\text{since } \boldsymbol{g}_1^T \boldsymbol{d}_0 = 0 \text{ from Equation 2.24b}\right) \end{aligned} \tag{2.27b}$$

Proof of ii) We need to prove that $\boldsymbol{g}_2^T \boldsymbol{d}_0 = 0$ and $\boldsymbol{g}_2^T \boldsymbol{d}_1 = 0$.

$$\begin{aligned} \boldsymbol{g}_2^T \boldsymbol{d}_0 &= \left(\boldsymbol{g}_1 + s_1 \boldsymbol{Q}\boldsymbol{d}_1\right)^T \boldsymbol{d}_0 \\ &= \boldsymbol{g}_1^T \boldsymbol{d}_0 + s_1 \boldsymbol{d}_1^T \boldsymbol{Q}\boldsymbol{d}_0 \\ &= 0 \left(\text{from Equation 2.24b and since } \boldsymbol{d}_1^T \boldsymbol{Q}\boldsymbol{d}_0 = 0\right) \end{aligned} \tag{2.28a}$$

$$\begin{aligned} \boldsymbol{g}_2^T \boldsymbol{d}_1 &= \boldsymbol{g}_2^T \left(-\boldsymbol{g}_1 + \beta_0 \boldsymbol{d}_0\right) \\ &= -\boldsymbol{g}_2^T \boldsymbol{g}_1 + \beta_0 \boldsymbol{g}_2^T \boldsymbol{d}_0 \\ &= 0 \quad \left(\boldsymbol{g}_2^T \boldsymbol{g}_1 = 0 \text{ from Equation 2.27b,}\right. \\ &\qquad \left.\boldsymbol{g}_2^T \boldsymbol{d}_0 = 0 \text{ from Equation 2.28a}\right) \end{aligned} \tag{2.28b}$$

Now we are ready to proceed to the k^{th} iteration.

Iteration k

$$\boldsymbol{g}_k = \nabla f\left(\boldsymbol{x}_k\right) = \boldsymbol{Q}\boldsymbol{x}_k + \boldsymbol{b} = \boldsymbol{g}_{k-1} + s_{k-1}\boldsymbol{Q}\boldsymbol{d}_{k-1} \tag{2.29}$$

The direction $\boldsymbol{d}_k$ is obtained by imposing the $\boldsymbol{Q}$-conjugacy requirement with $\boldsymbol{d}_{k-1}$. Similar to Equations (2.22) and (2.23) of the first iteration and (2.25d) and (2.25f) of the second iteration, one has:

$$\boldsymbol{d}_k = -\boldsymbol{g}_k + \beta_{k-1}\boldsymbol{d}_{k-1} \text{ with } \beta_{k-1} = \frac{\boldsymbol{d}_{k-1}^T\boldsymbol{Q}\boldsymbol{g}_k}{\boldsymbol{d}_{k-1}^T\boldsymbol{Q}\boldsymbol{d}_{k-1}} \tag{2.30}$$

and the update is:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + s_k\boldsymbol{d}_k, \qquad \text{with} \quad s_k = -\frac{\boldsymbol{d}_k^T\boldsymbol{g}_k}{\boldsymbol{d}_k^T\boldsymbol{Q}\boldsymbol{d}_k} = -\frac{\boldsymbol{g}_k^T\boldsymbol{d}_k}{\boldsymbol{d}_k^T\boldsymbol{Q}\boldsymbol{d}_k} \tag{2.31}$$

The iterative process reaches the optimum $\boldsymbol{x}^*$ in exactly n iterations for an n-dimensional quadratic function (Figure 2.3).

Some salient features of CG method
The CG method is characterized by:

$$\text{(i)} \quad \boldsymbol{g}_{k+1}^T\boldsymbol{g}_k = 0 \tag{2.32}$$

$$\text{(ii)} \quad \boldsymbol{g}_{k+1}^T\boldsymbol{d}_i = 0, i < k+1 \tag{2.33}$$

The first characteristic implies that $\boldsymbol{g}_{k+1}$ is orthogonal to $\boldsymbol{g}_k$, for all $k = 0,1,\ldots,n-1$ and the second one states that $\boldsymbol{g}_{k+1}$ is orthogonal to $\boldsymbol{d}_i$ for all $i < k+1$. This is shown to be true for $k = 0$ and 1 – see Equations (2.24a,b) for $k = 0$ and Equations (2.26a,b) for $k = 1$ and is indeed true for $k > 1$.

Simplified formula for step size s_k
Equation (2.33) can be used to simplify the expression for s_k in Equation (2.31).

$$s_k = -\frac{\boldsymbol{g}_k^T\boldsymbol{d}_k}{\boldsymbol{d}_k^T\boldsymbol{Q}\boldsymbol{d}_k} = -\frac{\boldsymbol{g}_k^T\left(-\boldsymbol{g}_k + s_{k-1}\boldsymbol{d}_{k-1}\right)}{\boldsymbol{d}_k^T\boldsymbol{Q}\boldsymbol{d}_k} \text{ (from Equation 2.30)}$$

$$= \frac{\boldsymbol{g}_k^T\boldsymbol{g}_k}{\boldsymbol{d}_k^T\boldsymbol{Q}\boldsymbol{d}_k} - \frac{s_{k-1}\boldsymbol{g}_k^T\boldsymbol{d}_{k-1}}{\boldsymbol{d}_k^T\boldsymbol{Q}\boldsymbol{d}_k}$$

$$= \frac{\boldsymbol{g}_k^T\boldsymbol{g}_k}{\boldsymbol{d}_k^T\boldsymbol{Q}\boldsymbol{d}_k}\left(\boldsymbol{g}_k^T\boldsymbol{d}_{k-1} = 0 \text{ from Equation } 2.33\right) \tag{2 .34}$$

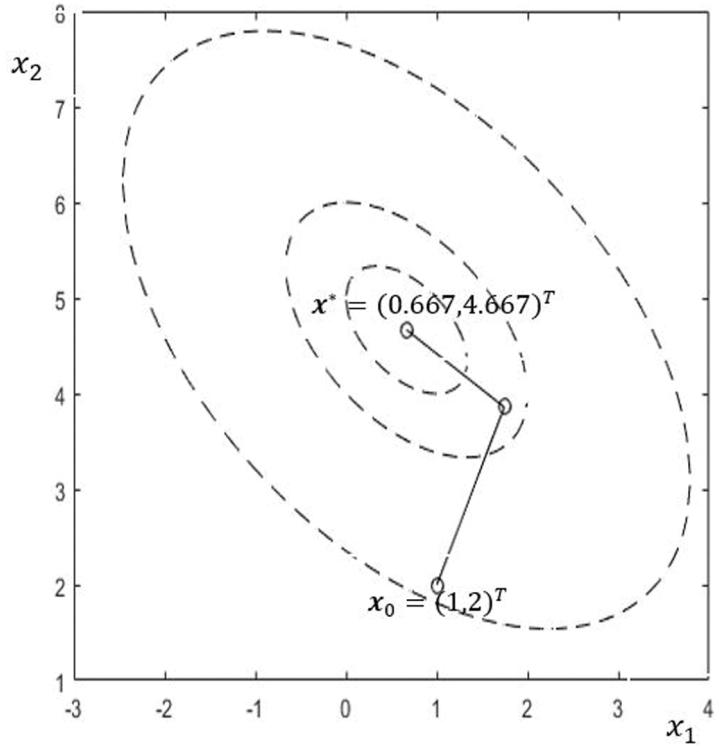


FIGURE 2.3 CG method and convergence of a quadratic function; $\boldsymbol{x}_0 = (1,2)^T$ is the starting point, $\boldsymbol{x}^* = (0.667, 4.667)^T$ is the optimum point; $f(x_1, x_2) = (x_1 - 3)^2 + (x_2 - 5)^2 + x_1 x_2$; optimum realized in $n = 2$ iterations.

The parameter β_{k-1} in Equation (2.30) may be simplified by Equation (2.32). Noting that

$$\boldsymbol{g}_k = \boldsymbol{g}_{k-1} + s_{k-1}\boldsymbol{Q}\boldsymbol{d}_{k-1} \tag{2.35a}$$

one has:

$$\boldsymbol{Q}\boldsymbol{d}_{k-1} = s_{k-1}^{-1}\left(\boldsymbol{g}_k - \boldsymbol{g}_{k-1}\right) \tag{2.35b}$$

Substituting in Equation (2.30) gives:

$$\beta_{k-1} = \frac{\boldsymbol{d}_{k-1}^T \boldsymbol{Q}\boldsymbol{g}_k}{\boldsymbol{d}_{k-1}^T \boldsymbol{Q}\boldsymbol{d}_{k-1}} = \frac{\boldsymbol{g}_k^T \boldsymbol{Q}\boldsymbol{d}_{k-1}}{\boldsymbol{d}_{k-1}^T \boldsymbol{Q}\boldsymbol{d}_{k-1}} = \frac{s_{k-1}^{-1}\boldsymbol{g}_k^T\left(\boldsymbol{g}_k - \boldsymbol{g}_{k-1}\right)}{\boldsymbol{d}_{k-1}^T \boldsymbol{Q}\boldsymbol{d}_{k-1}}$$

$$= \left(\frac{\boldsymbol{d}_{k-1}^T \boldsymbol{Q}\boldsymbol{d}_{k-1}}{\boldsymbol{g}_{k-1}^T \boldsymbol{g}_{k-1}}\right)\frac{\boldsymbol{g}_k^T\left(\boldsymbol{g}_k - \boldsymbol{g}_{k-1}\right)}{\boldsymbol{d}_{k-1}^T \boldsymbol{Q}\boldsymbol{d}_{k-1}} \left(\text{substituting for } s_{k-1} \text{ from Equation 2.34}\right)$$

$$= \frac{\boldsymbol{g}_k^T \boldsymbol{g}_k}{\boldsymbol{g}_{k-1}^T \boldsymbol{g}_{k-1}} \left(\text{since } \boldsymbol{g}_k^T \boldsymbol{d}_{k-1} = 0 \text{ from Equation 2.32}\right) \tag{2.36}$$

CG method for a non-quadratic function

For a non-quadratic function $f(\boldsymbol{x})$, the equality $\boldsymbol{g}_k = (\boldsymbol{Q}\boldsymbol{x}_k + \boldsymbol{b})$ is not applicable. It is the gradient $\nabla f(\boldsymbol{x}_k) = \boldsymbol{g}_k$ to be used in Equations (2.35) and (2.36) to get the parameter β_{k-1} and the step size s_k at the k^{th} iteration. Further, the matrix $\boldsymbol{Q}$ needs to be replaced by the Hessian matrix = $\boldsymbol{H} = \left[\frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j}\right]_{n \times n}$. However, to avoid computing the $\boldsymbol{H}$ matrix, the step size s_k may be obtained at the k^{th} iteration by a line search – minimizing $f(\boldsymbol{x}_k + s_k \boldsymbol{d}_k)$ with respect to s_k. Similarly, the simplified formula $\beta_{k-1} = \frac{\boldsymbol{g}_k^T \boldsymbol{g}_k}{\boldsymbol{g}_{k-1}^T \boldsymbol{g}_{k-1}}$ (in Equation 2.36 which is applicable for quadratic functions) is suggested for non-quadratic functions also by Fletcher and Reeves (1964). Table 2.1 gives the algorithmic steps for the CG method.

Realizing the optimum $\boldsymbol{x}^*$ for non-quadratic functions may take more than n iterations. Figures 2.4a–b show the optimization result by CG for a non-quadratic

TABLE 2.1
Algorithm of CG Method

Consider an n-dimensional objective function $f(\boldsymbol{x})$ with $\boldsymbol{x} \in \mathbb{R}^n$.
Step 1. Initiate iterations with $k = 0$. Start with initial point $\boldsymbol{x}_0$. Set $\boldsymbol{g}_0 = \nabla f(\boldsymbol{x}_0)$.
Let the first conjugate direction be $\boldsymbol{d}_0 = -\boldsymbol{g}_0$. Fix convergence parameters
ε_g and ε_f for the gradient and the objective function respectively.

Start iterations; $k = 0, 1, \ldots, n-1$.

Step 2. If $\|\boldsymbol{g}_k\| \le \varepsilon_g$, set the optimum $\boldsymbol{x}^* = \boldsymbol{x}_k$ and stop the iterative process.

Otherwise, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + s_k \boldsymbol{d}_k$.
If $k = 0$, $\boldsymbol{d}_0 = -\boldsymbol{g}_0$, Otherwise,

$\boldsymbol{d}_k = -\boldsymbol{g}_k + \beta_{k-1} \boldsymbol{d}_{k-1}$ with $\beta_{k-1} = \frac{\boldsymbol{g}_k^T \boldsymbol{g}_k}{\boldsymbol{g}_{k-1}^T \boldsymbol{g}_{k-1}}$ (Equation 2.36).

Step 3. Find s_k by line search by minimizing $f(\boldsymbol{x}_k + s_k \boldsymbol{d}_k)$ with respect to s_k and locate $\boldsymbol{x}_{k+1}$.

If $|f(\boldsymbol{x}_{k+1})| \le \varepsilon_f$, set the optimum $\boldsymbol{x}^* = \boldsymbol{x}_{k+1}$ and stop the iterative process.

Otherwise, set $\boldsymbol{g}_{k+1} = \nabla f(\boldsymbol{x}_{k+1})$.
Step 4. If $k < n-1$, set $k = k+1$ and return to step 2.

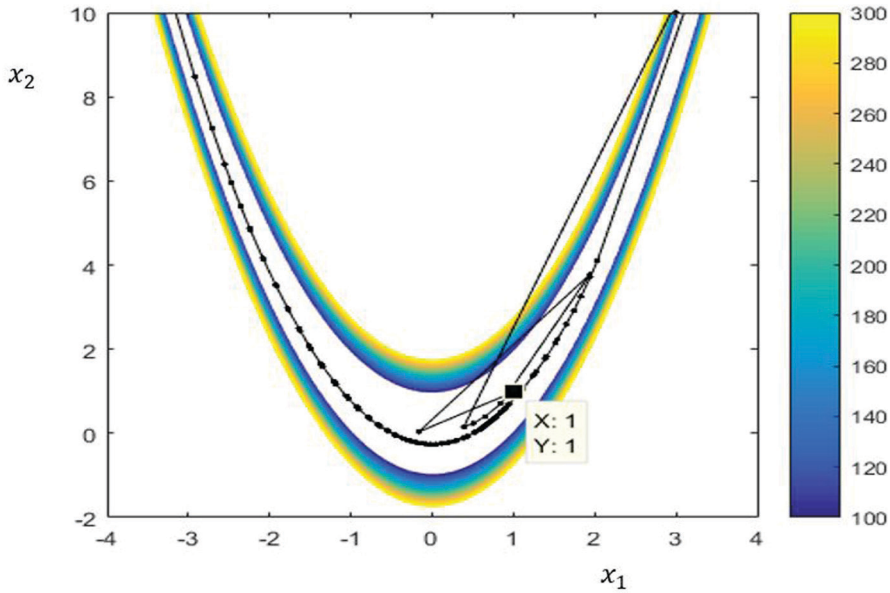


FIGURE 2.4a Conjugate gradient method applied to Rosenbrock function $f(x_1,x_2)=100\left(x_1^2-x_2\right)^2+\left(x_1-1\right)^2$, $\boldsymbol{x}_0=(3,10)^T$, $\boldsymbol{x}^*=(1,1)^T$; distribution of iterations in parameter space (convergence in 175 iterations).

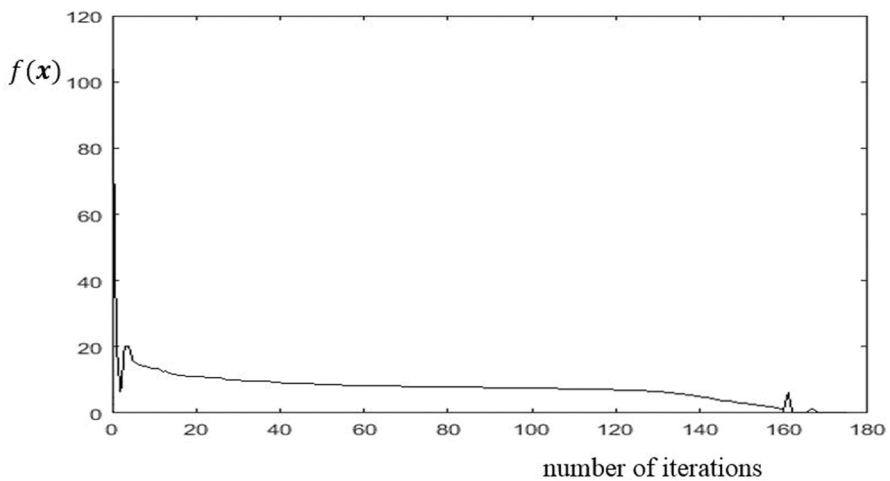


FIGURE 2.4b Conjugate gradient method applied to Rosenbrock function $f(x_1,x_2)=100\left(x_1^2-x_2\right)^2+\left(x_1-1\right)^2$, $\boldsymbol{x}_0=(3,10)^T$, $\boldsymbol{x}^*=(1,1)^T$; evolution of objective function with iterations (attaining a minimum value of 2.52E-13 at the end of 175 iterations).

function $f(\boldsymbol{x}) = 100\left(x_1^2 - x_2\right)^2 + \left(x_1 - 1\right)^2$, known as Rosenbrock function, with $\boldsymbol{x} = \left(x_1, x_2\right)^T$.

CG method and application to a system of linear equations

While the CG method as applied to non-linear optimization problems has been attributed to Fletcher and Reeves (1964), Hestenes and Stiefel (1952) originally proposed the algorithm as an effective approach to solve large scale symmetric, positive-definite linear system of equations. For numerical purposes, the method has been shown to be superior to Gaussian elimination. Finding a solution to a system of equations $\boldsymbol{Ax} = \boldsymbol{b}$ is equivalent to searching for the minimum of the quadratic function $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Ax} - \boldsymbol{b}^T\boldsymbol{x}$. Hence the algorithm in Table 2.1 may be used to solve $\boldsymbol{Ax} = \boldsymbol{b}$.

Preconditioned CG

Having a fast matrix solver by speeding up the convergence rate is a necessity particularly for large sparse linear systems. Key for fast convergence by gradient methods is to have a low condition number of the matrix $\boldsymbol{A}$; see the discussion in the earlier Section 2.2.1. Pre-conditioning of the matrix $\boldsymbol{A}$ is one useful technique (Golub and Van Loan 1996, Benzi 2002, Watkins 2010, Datta 2010) utilized in almost all commercially available solvers for better convergence. Pre-conditioning yields a better distribution of the system eigenvalues, so that condition number of $\boldsymbol{A}$ is driven closer to 1.

If $\boldsymbol{M}$ is an invertible matrix and the condition number of $\boldsymbol{M}^{-1}\boldsymbol{A}$ is smaller than that of $\boldsymbol{A}$, then $\boldsymbol{M}$ is a possible choice for a pre-conditioner. Since $\boldsymbol{A}$ is symmetric and positive definite, $\boldsymbol{M}$ also needs to be symmetric and positive definite. One choice is to select $\boldsymbol{M}$ as a diagonal matrix with the diagonal elements of $\boldsymbol{A}$ as its non-zero entries, i.e.:

$$\begin{aligned} M_{ii} &= A_{ii} \\ &= 0, \text{ if } i \neq j \end{aligned} \tag{2.37}$$

This is known as the Jacobi pre-conditioner. Another choice is given by $\boldsymbol{M}^{-1} = \boldsymbol{LL}^T$ where $\boldsymbol{L}$ is a lower triangular matrix obtained through Cholesky decomposition of $\boldsymbol{A}$ (Meijerink and van der Vorst 1977, Kershaw 1978, Chan and van der Vorst 1997). This ensures that $\boldsymbol{M}$ is symmetric and positive definite. However, $\boldsymbol{L}$ loses sparsity when compared to $\boldsymbol{A}$. To compensate for this deficiency, $\boldsymbol{L}$ is modified such that if an off-diagonal term $A_{ij,i\neq j} = 0$, $L_{ij,i\neq j}$ is also set to zero. It amounts to having an

incomplete Cholesky (IC) decomposition (Golub and Van Loan 1996) of $\boldsymbol{A}$, and $\boldsymbol{L}$ is ensured to have at least the same sparsity as $\boldsymbol{A}$.

Note that for any choice of $\boldsymbol{M}$ with $\boldsymbol{M}^{-1} = \boldsymbol{L}\boldsymbol{L}^T$, an equivalent system may be constructed as follows:

$$\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{M}^{-1}\boldsymbol{b}$$

$$\Rightarrow \boldsymbol{L}\boldsymbol{L}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{L}\boldsymbol{L}^T\boldsymbol{b}$$

$$\Rightarrow \left[\boldsymbol{L}^T\boldsymbol{A}\boldsymbol{L}\right]\left\{\boldsymbol{L}^{-1}\boldsymbol{x}\right\} = \left\{\boldsymbol{L}^T\boldsymbol{b}\right\}$$

$$\Rightarrow \hat{\boldsymbol{A}}\hat{\boldsymbol{x}} = \hat{\boldsymbol{b}} \tag{2.38}$$

where $\hat{\boldsymbol{A}} = \boldsymbol{L}^T\boldsymbol{A}\boldsymbol{L}$ and $\hat{\boldsymbol{b}} = \boldsymbol{L}^T\boldsymbol{b}$. The new solution vector is $\hat{\boldsymbol{x}} = \boldsymbol{L}^{-1}\boldsymbol{x}$. $\hat{\boldsymbol{A}}$ remains symmetric positive definite since $\hat{\boldsymbol{x}}^T\hat{\boldsymbol{A}}\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}^T\left[\boldsymbol{L}^T\boldsymbol{A}\boldsymbol{L}\right]\hat{\boldsymbol{x}} = \left(\boldsymbol{L}\hat{\boldsymbol{x}}\right)^T\boldsymbol{A}\left(\boldsymbol{L}\hat{\boldsymbol{x}}\right) = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 0$.

The new algorithm closely follows the iterative steps of the original CG method. Though the new variables $\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{g}}_k$ and $\hat{\boldsymbol{d}}_k$ respectively replace the original variables $\boldsymbol{x}_k, \boldsymbol{g}_k$ and $\boldsymbol{d}_k$ in the formulae, it is computationally convenient to proceed iteratively with the new steps stated in terms of the old ones.

At the k^{th} iteration, the residual is $\hat{\boldsymbol{g}}_k = \hat{\boldsymbol{b}} - \hat{\boldsymbol{A}}\hat{\boldsymbol{x}}_k$. It is related to the residual $\boldsymbol{g}_k$ as:

$$\hat{\boldsymbol{g}}_k = \hat{\boldsymbol{b}} - \hat{\boldsymbol{A}}\hat{\boldsymbol{x}}_k = \boldsymbol{L}^T\boldsymbol{b} - \boldsymbol{L}^T\boldsymbol{A}\boldsymbol{L}\,\boldsymbol{L}^{-1}\boldsymbol{x}_k = \boldsymbol{L}^T\left(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k\right) = \boldsymbol{L}^T\boldsymbol{g}_k \tag{2.39}$$

The update is given by:

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_k + \hat{s}_k\hat{\boldsymbol{d}}_k$$

$$= \boldsymbol{L}^{-1}\boldsymbol{x}_k + \hat{s}_k\hat{\boldsymbol{d}}_k \tag{2.40a}$$

If the new direction $\hat{\boldsymbol{d}}_k$ is defined as $\boldsymbol{L}^{-1}\boldsymbol{d}_k$, then $\hat{\boldsymbol{x}}_{k+1}$ simplifies to:

$$\hat{\boldsymbol{x}}_{k+1} = \boldsymbol{x}_k + \hat{s}_k\boldsymbol{d}_k \tag{2.40b}$$

Following Equation (2.30), the new direction $\hat{\boldsymbol{d}}_k$ for $k > 0$ can also be written as:

$$\hat{\boldsymbol{d}}_k = -\hat{\boldsymbol{g}}_k + \hat{\beta}_{k-1}\hat{\boldsymbol{d}}_{k-1}$$

$$\Rightarrow \boldsymbol{L}^{-1}\boldsymbol{d}_k = -\boldsymbol{L}^T\boldsymbol{g}_k + \hat{\beta}_{k-1}\boldsymbol{L}^{-1}\boldsymbol{d}_{k-1}$$

$$\Rightarrow \boldsymbol{d}_k = -\boldsymbol{L}\boldsymbol{L}^T\boldsymbol{g}_k + \hat{\beta}_{k-1}\boldsymbol{d}_{k-1}$$

$$\Rightarrow \boldsymbol{d}_k = -\boldsymbol{M}^{-1}\boldsymbol{g}_k + \hat{\beta}_{k-1}\boldsymbol{d}_{k-1} \tag{2.41}$$

The parameter $\hat{\beta}_{k-1}$ follows from Equation (2.36):

$$\hat{\beta}_{k-1} = \frac{\hat{\boldsymbol{g}}_k^T \hat{\boldsymbol{g}}_k}{\hat{\boldsymbol{g}}_{k-1}^T \hat{\boldsymbol{g}}_{k-1}}$$

$$= \frac{\left(\boldsymbol{L}^T \boldsymbol{g}_k\right)^T \left(\boldsymbol{L}^T \boldsymbol{g}_k\right)}{\left(\boldsymbol{L}^T \boldsymbol{g}_{k-1}\right)^T \left(\boldsymbol{L}^T \boldsymbol{g}_{k-1}\right)} = \frac{\boldsymbol{g}_1^T \boldsymbol{M}^{-1} \boldsymbol{g}_k}{\boldsymbol{g}_{k-1}^T \boldsymbol{M}^{-1} \boldsymbol{g}_{k-1}} \tag{2.42}$$

Note that for $k = 0$, the starting direction $\hat{\boldsymbol{d}}_0 = \hat{\boldsymbol{g}}_0$. This gives:

$$\hat{\boldsymbol{d}}_0 = \boldsymbol{L}^{-1}\boldsymbol{d}_0 = \hat{\boldsymbol{g}}_0 = \boldsymbol{L}^T \boldsymbol{g}_0$$

$$\Rightarrow \boldsymbol{d}_0 = \boldsymbol{L}\boldsymbol{L}^T \boldsymbol{g}_0 = \boldsymbol{M}^{-1}\boldsymbol{g}_0 \tag{2.43}$$

The step size $\hat{s}_k$ in Equation (2.40) is obtained as:

$$\hat{s}_k = \frac{\hat{\boldsymbol{g}}_k^T \hat{\boldsymbol{g}}_k}{\hat{\boldsymbol{d}}_k^T \hat{\boldsymbol{A}} \hat{\boldsymbol{d}}_k} = \frac{\left(\boldsymbol{L}^T \boldsymbol{g}_k\right)^T \boldsymbol{L}^T \boldsymbol{g}_k}{\left(\boldsymbol{L}^{-1}\boldsymbol{d}_k\right)^T \boldsymbol{L}^T \boldsymbol{A}\boldsymbol{L}\boldsymbol{L}^{-1}\boldsymbol{d}_k}$$

$$= \frac{\boldsymbol{g}_k^T \boldsymbol{L}\boldsymbol{L}^T \boldsymbol{g}_k}{\boldsymbol{d}_k^T \boldsymbol{L}^{-T} \boldsymbol{L}^T \boldsymbol{A}\boldsymbol{L}\boldsymbol{L}^{-1}\boldsymbol{d}_k} = \frac{\boldsymbol{g}_k^T \boldsymbol{M}^{-1} \boldsymbol{g}_k}{\boldsymbol{d}_k^T \boldsymbol{A}\boldsymbol{d}_k} \tag{2.44}$$

For the next iteration, the residual is updated as in Equation (2.29):

$$\hat{\boldsymbol{g}}_{k+1} = \hat{\boldsymbol{g}}_k - \hat{s}_k \hat{\boldsymbol{A}} \hat{\boldsymbol{d}}_k$$

$$\Rightarrow \boldsymbol{L}^T \boldsymbol{g}_{k+1} = \boldsymbol{L}^T \boldsymbol{g}_k - \hat{s}_k \boldsymbol{L}^T \boldsymbol{A}\boldsymbol{L}\boldsymbol{L}^{-1}\boldsymbol{d}_k$$

$$\Rightarrow \boldsymbol{g}_{k+1} = \boldsymbol{g}_k - \hat{s}_k \boldsymbol{A}\boldsymbol{d}_k \tag{2.45}$$

Table 2.2 details the algorithmic steps for the pre-conditioned CG method.

CG method with pre-conditioning is of significance in application to linear systems arising from finite difference/finite element discretization of boundary-value problems. The example below illustrates an application to a heat transfer problem.

TABLE 2.2
Algorithm of the Preconditioned CG Method

Consider a system of n linear equations $\boldsymbol{Ax} = \boldsymbol{b}$. Choose a pre-conditioner $\boldsymbol{M}$.
Step 1. Initiate iterations with $k = 0$ .Start with an initial point $\boldsymbol{x}_0$. Set $\boldsymbol{g}_0 = \boldsymbol{b} - \boldsymbol{Ax}_0$.
Let the first conjugate direction be $\boldsymbol{d}_0 = \boldsymbol{M}^{-1}\boldsymbol{g}_0$ (Equation 2.43).
Fix convergence parameter ε_g with respect to the residual $\boldsymbol{g} = \boldsymbol{b} - \boldsymbol{Ax}$.
Start iterations; $k = 0,1,\ldots,n-1$.
Step 2. If $\|\boldsymbol{g}_k\| \le \varepsilon_g$, set the solution $\boldsymbol{x}^* = \boldsymbol{x}_k$ and stop the iterative process.
Otherwise, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \hat{s}_k \boldsymbol{d}_k$
Step 3. Obtain the step size $\hat{s}_k = \dfrac{\boldsymbol{g}_k^T \boldsymbol{M}^{-1} \boldsymbol{g}_k}{\boldsymbol{d}_k^T \boldsymbol{A} \boldsymbol{d}_k}$ (Equation 2.44)
where $\boldsymbol{d}_k = -\boldsymbol{M}^{-1}\boldsymbol{g}_k + \hat{\beta}_{k-1}\boldsymbol{d}_{k-1}$.
The parameter $\hat{\beta}_{k-1}$ is obtained from Equation (2.42): $\hat{\beta}_{k-1} = \dfrac{\boldsymbol{g}_1^T \boldsymbol{M}^{-1} \boldsymbol{g}_k}{\boldsymbol{g}_{k-1}^T \boldsymbol{M}^{-1} \boldsymbol{g}_{k-1}}$
Step 4. Knowing $\hat{s}_k$ and $\boldsymbol{d}_k$, find the update $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \hat{s}_k \boldsymbol{d}_k$.
Step 5. Evaluate the residual $\boldsymbol{g}_{k+1} = \boldsymbol{g}_k - \hat{s}_k \boldsymbol{A} \boldsymbol{d}_k$ (Equation 2.45)
Step 6. Set $k = k + 1$ and return to step 2.

Even though the resulting system of equations is of a very low dimension and may not warrant the use of the CG method, the example is still instructive.

Example 2.1. We consider a two-dimensional steady state heat flow problem and the heat flow is governed by the Poisson equation (Bathe 1996):

$$\frac{\partial}{\partial x}\left(k_x \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial u}{\partial y}\right) + Q = 0 \tag{2.46}$$

$u(x, y)$ is the field variable representing the temperature in the body occupying a region $\Omega \in \mathbb{R}^2$. $Q(x, y)$ is the heat source. k_x and k_y are the thermal conductivities of the material in x and y directions respectively. We wish to find the temperature distribution in the plate (Figure 2.5).

Solution. Let the body be isotropic and homogeneous with $k_x = k_y = k = 10$ watts/cm/Kelvin. Equation (2.46) then simplifies to:

$$k\Delta u + Q = 0 \tag{2.47}$$

$\Delta = \nabla^2 = \dfrac{\partial^2}{\partial x^2} + \dfrac{\partial^2}{\partial y^2}$ is the Laplacian operator. No heat sources are assumed i.e., Q =0.
The following temperature profile is applied at the boundary $y = l$.

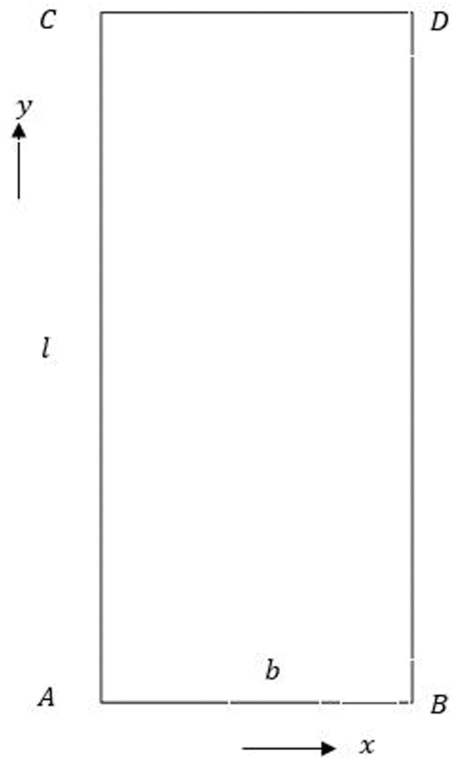


FIGURE 2.5 Steady-state heat flow problem; a rectangular plate ABCD of homogeneous and isotropic material – length l = 10 cm and width b = 5 cm.

$$u(x,l) = 100 \sin \frac{\pi x}{10} \tag{2.48}$$

This is a Dirichlet boundary condition. PDE (2.47) along with the BC in (2.48) forms an elliptic boundary value problem (BVP).[§] To find the solution to the BVP, we use

---

[§] Elliptic boundary value problem

A second-order partial differential operator A on a scalar valued function u of n variables $x_i, i = 1,2,\ldots,n$ may be expressed as:

$$Au = \sum_i \sum_j \frac{\partial}{\partial x_i}\left(a_{ij}(x) \frac{\partial u}{\partial x_j} \right) + \sum_j b_j \frac{\partial u}{\partial x_j} + cu = f \text{ in a domain } \mho \tag{i}$$

with suitable coefficients a_{ij}, b_j and c. A is elliptic if:

$$l(x,\xi) = \sum_i \sum_j a_{ij}(x) \xi_i \xi_j \neq 0, \forall\ \xi \neq 0 \tag{ii}$$

where l is a polynomial of order 2 in the components $\xi = \xi_i, i = 1,2,\ldots,n$. Equation (i) satisfying condition (ii) and followed by boundary conditions (conditions on the boundary $\partial\mho$) defines an elliptic boundary value problem.

the finite element method (FEM) and follow the procedure described in Section 1.5.1, Chapter 1. To this end, we first convert the strong form of the governing Equation (2.47) to a weak form:

$$\int_{\Omega}\left\{\frac{1}{2}\left(\nabla v.\nabla u\right)+Qv\right\}d\Omega=0 \tag{2.49}$$

v is the test function. Both u and v belong to the Sobolev space $H^1\left(\Omega\right)$ (Appendix B). The integral $\int_{\Omega}\frac{1}{2}(\nabla v.\nabla u)d\Omega$ in the first term is a symmetric bilinear form. The integral $\int_{\Omega}Qvd\Omega$ in the second term is a linear form. Seeking a solution by FEM, we discretize the domain Ω into non-overlapping triangular elements using n nodes. With a linear variation of temperature within an element, the element-wise shape functions which form a polynomial basis set are assumed to be:

$$Y_j^e\left(x,y\right)=a_j+b_jx+c_jy,\ j=1,2,3 \tag{2.50}$$

$a_j,b_j,c_j,j=1,2,3\in\mathbb{R}$. The superscript '$e$' stands for an element and the total number of elements in the model is denoted by N_e. The temperature distribution in a triangular element is:

$$u^e\left(x,y\right)=\sum_{j=1}^{3}Y_j^e\left(x,y\right)T_j^e \tag{2.51}$$

$T_j^e,j=12,3$ are the unknown nodal temperatures of the element. Substituting Equation (2.51) into the weak form (2.49) and integrating over the element domain Ω^e yield a set of discrete equations of the form (Bathe 1996):

$$\boldsymbol{K}^e\boldsymbol{T}^e=\boldsymbol{F}^e \tag{2.52}$$

$\boldsymbol{K}^e$ is a 3×3 symmetric matrix of the form (only the top half including the diagonal entries are shown):

$$\boldsymbol{K}^e=\frac{k}{4A_e}\begin{bmatrix}\left(b_1^2+c_1^2\right) & \left(b_1b_2+c_1c_2\right) & \left(b_1b_3+c_1c_3\right)\\ & \left(b_2^2+c_2^2\right) & \left(b_2b_3+c_2c_3\right)\\ & & \left(b_3^2+c_3^2\right)\end{bmatrix} \tag{2.53}$$

A_e is the area of the triangular element. If $\left(x_j,y_j\right)$, $j=1,2,3$ are the coordinates of the element vertices, the parameters b_j and c_j are given by

$b_1 = y_2 - y_3, b_2 = y_3 - y_1, b_3 = y_1 - y_2$ and $c_1 = x_3 - x_2,\ c_2 = x_1 - x_3,\ c_3 = x_2 - x_1$ (Liu and Quek 2003). The vector F^e on the RHS of Equation (2.52) is obtained as:

$$\boldsymbol{F}^e = \frac{\boldsymbol{Q}A_e}{3}\begin{pmatrix}1 & 1 & 1\end{pmatrix}^T \tag{2.54}$$

Since no heat sources are assumed, $\boldsymbol{F}^e$ is a zero vector. Assembly of the equations in (2.52) over all the elements by appropriately summing up entries pertaining to the *dof*s common to adjoining elements (Bathe 1996, Cook *et al.* 1989), one obtains the system of linear equations:

$$\hat{\boldsymbol{K}}\hat{\boldsymbol{T}} = \hat{\boldsymbol{F}} \tag{2.55}$$

$\hat{\boldsymbol{K}} \in \mathbb{R}^{n \times n}$ and $\hat{\boldsymbol{F}} \in \mathbb{R}^n$. $\hat{\boldsymbol{T}} = \left(T_1, T_2, \ldots, T_n\right)^T$ is the vector of the nodal temperatures. By imposing the specified BC on the top edge CD (Figure 2.5), the number of unknown nodal temperatures reduces to $m = n - \left(\text{nodes on edge CD}\right)$. Deleting from the matrix $\hat{\boldsymbol{K}}$ the rows and columns for the *dof*s corresponding to the Dirichlet BC and suitably modifying the RHS vector $\hat{\boldsymbol{F}}$, one obtains the final set of equations as:

$$\boldsymbol{K}\boldsymbol{T} = \boldsymbol{F} \tag{2.56}$$

$\boldsymbol{K} \in \mathbb{R}^{m \times m}$, $\boldsymbol{F} \in \mathbb{R}^m$ and $\boldsymbol{T} = \mathbb{R}^m$. The CG method along with pre-conditioning strategy is applied to solve these equations. Solution is obtained with two types of pre-conditioners – a) Jacobi pre-conditioner and b) pre-conditioner by IC decomposition of $\boldsymbol{K}$. Two FE models (Figure 2.6) of the plate – one with a coarse mesh and another with a relatively refined one – are employed to demonstrate the performance of the two pre-conditioners. While the temperature distribution presently turns out to be the same irrespective of the preconditioner (Figure 2.7), the one based on IC decomposition shows superior performance vis-à-vis the Jacobi pre-conditioner as judged by the execution time (Table 2.3). The Jacobi pre-conditioner is generally known to have poor convergence rate with increase in model size (Augarde *et al.* 2006)]. For the two model sizes in the example, solution is also realized by the CG method without pre-conditioning.

■

Before we close this section, it is pertinent to take note of an upsurge of a new class of domain decomposition pre-conditioners (Smith et al. 1996, Oliveira and Sorensen 1997, Quarteroni and Valli 1999, Benzi 2002). They are suitable for parallel computing that involves large-scale simulations. These pre-conditioners obviously avoid a full assembly of the system matrices. Element based pre-conditioners (Hughes *et al.* 1983, Tezduyar and Liou 1989, Hughes and Ferencz 1988) belong to this category and together with the conjugate gradient method are, for instance, shown to be useful for solving problems of solid and fluid mechanics. For application of pre-conditioned CG to equations with unsymmetric coefficient matrices, interested readers may refer to Fletcher (1976), Concus and Golub (1976) and Makinson and Shah (1986).

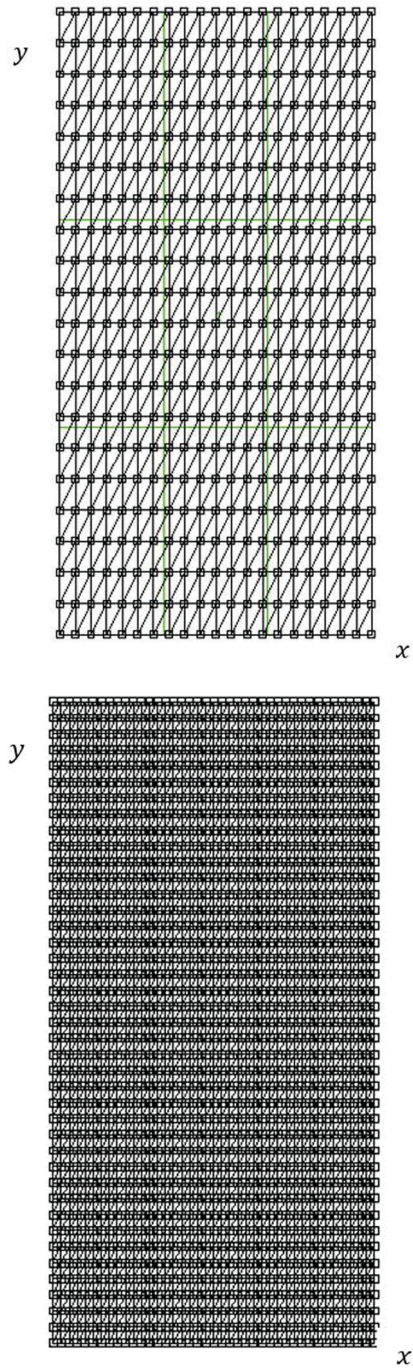


FIGURE 2.6 Application of CG method, steady state heat flow problem: (a) FE model with 441 nodes and 800 elements and (b) FE model with 1681 nodes and 3200 elements.

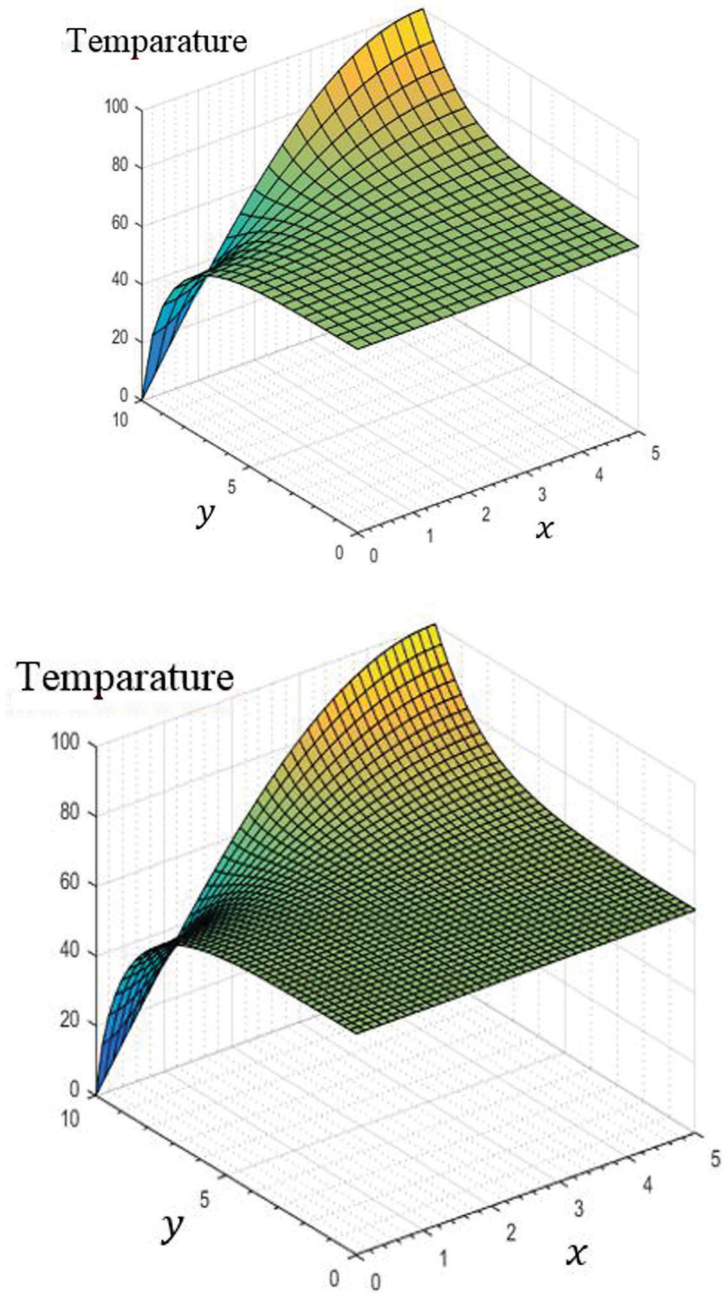


FIGURE 2.7 Solution to steady-state heat flow problem by CG method with Jacobi-preconditioning: (a) for the FE model in Figure 2.6a and (b) for the FE model in Figure 2.6b.

TABLE 2.3
Solution to Steady-state Heat Flow Problem – Comparison of Execution Time by CG Method with and without Pre-conditioners (on laptop version I7 with 8 GB RAM)

Method	Nodes = 21 × 21 Elements = 800 (Figure 2.6a)	Nodes = 41 × 41 Elements = 3200 (Figure 2.6b)
CG with no pre-conditioner	2.5 s	93.5 s
CG with Jacobi pre-conditioner, *M*	2.2 s	91.4 s
CG with pre-conditioner, *M* by IC decomposition	1.3 s	42.6 s

2.2.3 Newton's Method

For a function $f(\boldsymbol{x}) \in C^2$, a quadratic approximation $\hat{f}(\boldsymbol{x})$ in a neighbourhood of $\hat{\boldsymbol{x}}$ (Section 1.2, Chapter 1) is:

$$f(\boldsymbol{x}) \cong \hat{f}(\boldsymbol{x}) = f(\hat{\boldsymbol{x}}) + \nabla f(\hat{\boldsymbol{x}})^T (\boldsymbol{x} - \hat{\boldsymbol{x}}) + \frac{1}{2}(\boldsymbol{x} - \hat{\boldsymbol{x}})^T \boldsymbol{H}(\hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}}) \tag{2.57}$$

It follows that the gradient $\nabla f(\boldsymbol{x})$ approximates to:

$$\nabla f(\boldsymbol{x}) \cong \nabla f(\hat{\boldsymbol{x}}) + \boldsymbol{H}(\hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}}) \tag{2.58}$$

Imposing the condition $\nabla f(\boldsymbol{x}) = 0$ at the k^{th} iteration, one gets from the last equation the update for Newton's method as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{H}(\boldsymbol{x}_k)^{-1} \nabla f(\boldsymbol{x}_k) \tag{2.59}$$

Unlike the steepest descent and conjugate gradient methods which are of first order requiring only the gradient at each iteration, Newton's method is a second order one using both the gradient and the Hessian matrix. When $f(\boldsymbol{x})$ is convex and twice differentiable, Newton's method is quadratically convergent as is the case with the Newton-Raphson method. $-\boldsymbol{H}(\boldsymbol{x}_k)^{-1} \nabla f(\boldsymbol{x}_k)$ on the RHS of Equation (2.58) may be viewed as the new direction $\boldsymbol{d}_k$ at the k^{th} iteration. Figure 2.8 shows that Newton's method converges in exactly a single iteration for the quadratic function $f(\boldsymbol{x}) = (x_1 - 3)^2 + (x_2 - 5)^2 + x_1 x_2$. Steepest descent method took ten iterations

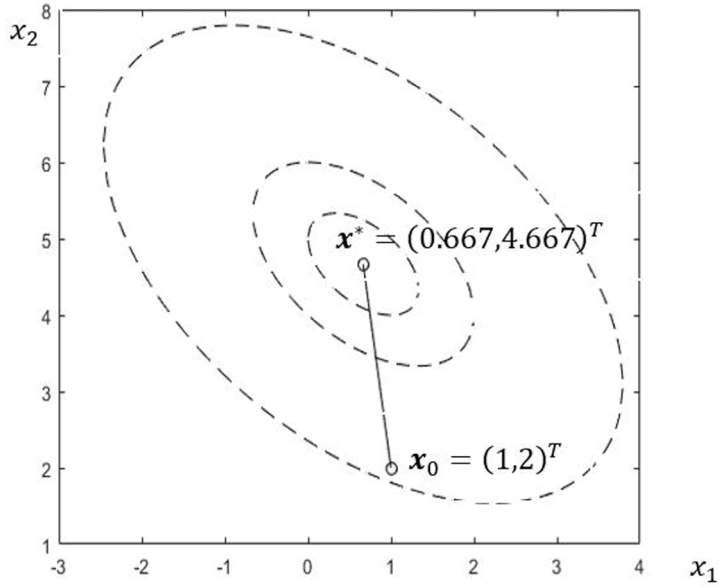


FIGURE 2.8 Newton's method and convergence of the quadratic function $f(x_1, x_2) = (x_2 - 5)^2 + (x_2 - 5)^2 + x_1 x_2$; $\boldsymbol{x}_0 = (1,2)^T$; $\boldsymbol{x}^* = (0.667,4.667)^T$.

(Figure 2.1b) and the conjugate gradient method two iterations (Figure 2.3) for convergence of the same quadratic function.

While handling non-quadratic functions, Newton's method solves a locally quadratic function at each iteration. For better convergence, one modification may be to have the update along the new descent direction $\boldsymbol{d}_k = -\boldsymbol{H}\left(\boldsymbol{x}_k\right)^{-1} \nabla f\left(\boldsymbol{x}_k\right)$ with a step size s_k, i.e.:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + s_k \boldsymbol{d}_k \tag{2.60}$$

s_k may be obtained by a line search.

Newton's method, though efficient in handling quadratic functions, often fails to converge for non-quadratic functions. The main reason is the loss of positive definiteness of the $\boldsymbol{H}$ matrix – violation of the sufficient condition for a minimum – during the iterative process, thereby leading to either no solution or an unacceptable one. Computing at each iteration the Hessian matrix $\boldsymbol{H}$ and its inverse constitutes another major computational disadvantage especially when applied to large dimensional optimization problems and for problems with no explicitly defined objective functions. Many variants of Newton's method – known as quasi-Newton methods – aim to correct these drawbacks.

2.3 QUASI-NEWTON METHODS

The quasi-Newton methods use an approximation to the Hessian matrix or its inverse during the iterative process. These methods sequentially generate the matrix whilst trying to keep it symmetric and positive definite.

2.3.1 Davidon-Fletcher-Powell (DFP) Method

The DFP method has been developed by Davidon (1959) and Fletcher and Powell (1963). It is a quasi-Newton method and uses, at each iteration, an approximation to the inverse $\boldsymbol{H}_k^{-1}\left(=\boldsymbol{H}\left(\boldsymbol{x}_k\right)^{-1}\right)$ of the Hessian. Using Equation (2.60), we may write:

$$f\left(\boldsymbol{x}_{k+1}\right)=f\left(\boldsymbol{x}_k\right)-s_k\nabla f\left(\boldsymbol{x}_k\right)^T\boldsymbol{H}_k^{-1}\nabla f\left(\boldsymbol{x}_k\right) \tag{2.61}$$

A decrease in the function value is guaranteed if $\nabla f\left(\boldsymbol{x}_k\right)^T\boldsymbol{H}_k^{-1}\nabla f\left(\boldsymbol{x}_k\right)>0$, i.e. if $\boldsymbol{H}_k^{-1}$ is positive definite. Starting from an arbitrary symmetric positive definite $\mathbb{F}_0$, the inverse Hessian is approximated in later iterations by $\mathbb{F}_1$, $\mathbb{F}_2,\ldots,\mathbb{F}_k,\ldots$. That is, $\mathbb{F}_k\approx\boldsymbol{H}_k^{-1}$. The method starts with $\mathbb{F}_0=\boldsymbol{I}_{n\times n}$, an identity matrix. At the k^{th} iteration, $\mathbb{F}_{k+1}$ is generated by the DFP formula as:

$$\mathbb{F}_{k+1}=\mathbb{F}_k+\frac{\boldsymbol{p}_k\boldsymbol{p}_k^T}{\boldsymbol{p}_k^T\boldsymbol{r}_k}-\frac{\mathbb{F}_k\boldsymbol{r}_k\boldsymbol{r}_k^T\mathbb{F}_k}{\boldsymbol{r}_k^T\mathbb{F}_k\boldsymbol{r}_k} \tag{2.62}$$

where $\boldsymbol{p}_k=\boldsymbol{x}_{k+1}-\boldsymbol{x}_k$ and $\boldsymbol{r}_k=\nabla f\left(\boldsymbol{x}_{k+1}\right)-\nabla f\left(\boldsymbol{x}_k\right)=\boldsymbol{g}_{k+1}-\boldsymbol{g}_k$. In the second term on the RHS of Equation (2.62), the denominator is a scalar quantity. The numerator $\boldsymbol{p}_k\boldsymbol{p}_k^T$ is a symmetric matrix with rank one. $\dfrac{\boldsymbol{p}_k\boldsymbol{p}_k^T}{\boldsymbol{p}_k^T\boldsymbol{r}_k}$ may be referred to as a rank one correction to $\mathbb{F}_k$. Likewise, the third term $\dfrac{\mathbb{F}_k\boldsymbol{r}_k\boldsymbol{r}_k^T\mathbb{F}_k}{\boldsymbol{r}_k^T\mathbb{F}_k\boldsymbol{r}_k}=\dfrac{\left(\mathbb{F}_k\boldsymbol{r}_k\right)\left(\mathbb{F}_k\boldsymbol{r}_k\right)^T}{\boldsymbol{r}_k^T\mathbb{F}_k\boldsymbol{r}_k}$ adds another rank one correction and keeps $\mathbb{F}_{k+1}$ symmetric. The two terms together constitute a rank two correction to $\mathbb{F}_k$. Let us now show by induction that $\mathbb{F}_{k+1}$ is positive definite.

Proof: $\mathbb{F}_0$ is positive definite by hypothesis. Let us assume that $\mathbb{F}_k$ is positive definite and show that $\mathbb{F}_{k+1}$ is also positive definite.

At the k^{th} iteration, the direction $\boldsymbol{d}_k=-\mathbb{F}_k\nabla f\left(\boldsymbol{x}_k\right)=-\mathbb{F}_k\boldsymbol{g}_k$ with $\boldsymbol{d}_0=-\boldsymbol{g}_0$. The update is:

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+s_k\boldsymbol{d}_k=\boldsymbol{x}_k-s_k\mathbb{F}_k\boldsymbol{g}_k \tag{2.63}$$

The step size s_k is obtained by a line search, i.e. by minimizing $f\left(\boldsymbol{x}_k + s_k \boldsymbol{d}_k\right)$ with respect to s_k. This implies:

$$\nabla f\left(\boldsymbol{x}_{k+1}\right)^T \boldsymbol{d}_k = \boldsymbol{g}_{k+1}^T \boldsymbol{d}_k = 0 \tag{2.64}$$

If we now consider the DFP formula in Equation (2.62), the associated quadratic function is:

$$\boldsymbol{y}^T \mathbb{F}_{k+1} \boldsymbol{y} = \boldsymbol{y}^T \mathbb{F}_k \boldsymbol{y} + \boldsymbol{y}^T \frac{\boldsymbol{p}_k \boldsymbol{p}_k^T}{\boldsymbol{p}_k^T \boldsymbol{r}_k} \boldsymbol{y} - \boldsymbol{y}^T \frac{\mathbb{F}_k \boldsymbol{r}_k \boldsymbol{r}_k^T \mathbb{F}_k}{\boldsymbol{r}_k^T \mathbb{F}_k \boldsymbol{r}_k} \boldsymbol{y}, \boldsymbol{y} \in \mathbb{R}^n \tag{2.65}$$

The second term on the RHS of Equation (2.65) is:

$$\boldsymbol{y}^T \frac{\boldsymbol{p}_k \boldsymbol{p}_k^T}{\boldsymbol{p}_k^T \boldsymbol{r}_k} \boldsymbol{y} = \frac{\left(\boldsymbol{p}_k^T \boldsymbol{y}\right)^T \boldsymbol{p}_k^T \boldsymbol{y}}{\boldsymbol{p}_k^T \boldsymbol{r}_k} \tag{2.66}$$

The numerator in the last equation is a positive quantity. For the denominator $\boldsymbol{p}_k^T \boldsymbol{r}_k$, one has:

$$\begin{aligned} \boldsymbol{p}_k^T \boldsymbol{r}_k &= s_k \boldsymbol{d}_k^T \left(\boldsymbol{g}_{k+1} - \boldsymbol{g}_k\right) = s_k \boldsymbol{g}_k^T \mathbb{F}_k \boldsymbol{g}_k > 0 \\ &\left(\text{since } \boldsymbol{d}_k^T \boldsymbol{g}_{k+1} = 0 \text{ from Equation } 2.64\right) \end{aligned} \tag{2.67}$$

In getting the above result, the relation $\boldsymbol{d}_k = -\mathbb{F}_k \boldsymbol{g}_k$ is used along with the fact that $\mathbb{F}_k$ is positive definite. Thus, the first rank one correction term is positive definite. For the remaining two terms (first and third) in the RHS of Equation (2.65) combined together, the associated quadratic function is:

$$\begin{aligned} \boldsymbol{y}^T \mathbb{F}_k \boldsymbol{y} - \boldsymbol{y}^T \frac{\mathbb{F}_k \boldsymbol{r}_k \boldsymbol{r}_k^T \mathbb{F}_k}{\boldsymbol{r}_k^T \mathbb{F}_k \boldsymbol{r}_k} \boldsymbol{y} &= \boldsymbol{y}^T \mathbb{F}_k \boldsymbol{y} - \frac{\boldsymbol{y}^T \mathbb{F}_k \boldsymbol{r}_k \boldsymbol{r}_k^T \mathbb{F}_k \boldsymbol{y}}{\boldsymbol{r}_k^T \mathbb{F}_k \boldsymbol{r}_k} \\ &= \boldsymbol{y}^T \boldsymbol{L}_k \boldsymbol{L}_k^T \boldsymbol{y} - \frac{\boldsymbol{y}^T \boldsymbol{L}_k \boldsymbol{L}_k^T \boldsymbol{r}_k \boldsymbol{r}_k^T \boldsymbol{L}_k \boldsymbol{L}_k^T \boldsymbol{y}}{\boldsymbol{r}_k^T \boldsymbol{L}_k \boldsymbol{L}_k^T \boldsymbol{r}_k} \end{aligned} \tag{2.68}$$

(with $\mathbb{F}_k = \boldsymbol{L}_k \boldsymbol{L}_k^T$ by Cholesky decomposition)

With $\boldsymbol{u} = \boldsymbol{L}_k^T \boldsymbol{y}$ and $\boldsymbol{v} = \boldsymbol{L}_k^T \boldsymbol{r}_k$, Equation (2.68) is rewritten as:

$$\begin{aligned} \boldsymbol{y}^T \mathbb{F}_k \boldsymbol{y} - \boldsymbol{y}^T \frac{\mathbb{F}_k \boldsymbol{r}_k \boldsymbol{r}_k^T \mathbb{F}_k}{\boldsymbol{r}_k^T \mathbb{F}_k \boldsymbol{r}_k} \boldsymbol{y} &= \boldsymbol{u}^T \boldsymbol{u} - \frac{\left(\boldsymbol{u}^T \boldsymbol{v}\right)\left(\boldsymbol{v}^T \boldsymbol{u}\right)}{\boldsymbol{v}^T \boldsymbol{v}} \\ &= \frac{\left(\boldsymbol{u}^T \boldsymbol{u}\right)\left(\boldsymbol{v}^T \boldsymbol{v}\right) - \left(\boldsymbol{u}^T \boldsymbol{v}\right)\left(\boldsymbol{v}^T \boldsymbol{u}\right)}{\boldsymbol{v}^T \boldsymbol{v}} \end{aligned} \tag{2.69}$$

The denominator $\boldsymbol{v}^T\boldsymbol{v}$ on the extreme RHS of the last equation is positive. For the numerator, one has:

$$\left(\boldsymbol{u}^T\boldsymbol{u}\right)\left(\boldsymbol{v}^T\boldsymbol{v}\right)-\left(\boldsymbol{u}^T\boldsymbol{v}\right)\left(\boldsymbol{v}^T\boldsymbol{u}\right)=\left(\boldsymbol{u}^T\boldsymbol{u}\right)\left(\boldsymbol{v}^T\boldsymbol{v}\right)-\left(\boldsymbol{u}^T\boldsymbol{v}\right)^2$$

>0 (since $\left(\boldsymbol{u}^T\boldsymbol{v}\right)^2\le\left(\boldsymbol{u}^T\boldsymbol{u}\right)\left(\boldsymbol{v}^T\boldsymbol{v}\right)$ by Schwartz inequality**) (2.70)

Thus, $\mathbb{F}_{k+1}$ from the DFP formula is positive definite. The direction $\boldsymbol{d}_{k+1}=-\mathbb{F}_{k+1}\nabla f\left(\boldsymbol{x}_{k+1}\right)$ is indeed a descent direction. ♦

For a quadratic function $f\left(\boldsymbol{x}\right)=\frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{b}^T\boldsymbol{x}+c$ with $\boldsymbol{Q}$ the Hessian matrix $\boldsymbol{H}$ at any iteration, the DFP method leads to the exact inverse $\boldsymbol{H}^{-1}$ in n iterations where n is the problem dimension. It is a conjugate gradient method and converges to the optimum $\boldsymbol{x}^*$ after at most n steps. The search directions are $\boldsymbol{H}$-conjugate:

$$\boldsymbol{d}_{k+1}^T\boldsymbol{H}\boldsymbol{d}_j=0,0\le j<k+1 \tag{2.71}$$

To prove this, we also need to show that the matrices $\mathbb{F}_0,\mathbb{F}_1,\ldots$ generated by the DFP formula satisfy the following relation:

$$\mathbb{F}_{k+1}\boldsymbol{r}_j=\boldsymbol{p}_j,0\le j<k+1 \tag{2.72}$$

The last equation is similar to the following relation that the Hessian $\boldsymbol{H}$ of the quadratic function satisfies:

$$\boldsymbol{H}\left(\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right)=\boldsymbol{g}_{k+1}-\boldsymbol{g}_k\Rightarrow\boldsymbol{H}\boldsymbol{p}_j=\boldsymbol{r}_j \tag{2.73}$$

We concurrently prove the two requirements in Equations (2.71) and (2.72) by induction.

Proof for relations (2.71) and (2.72): Set $k=0$. From Equation (2.72), one has:

$$\mathbb{F}_1\boldsymbol{r}_0=\left[\boldsymbol{I}_{n\times n}+\frac{\boldsymbol{p}_0\boldsymbol{p}_0^T}{\boldsymbol{p}_0^T\boldsymbol{r}_0}-\frac{\mathbb{F}_0\boldsymbol{r}_0\boldsymbol{r}_0^T\mathbb{F}_0}{\boldsymbol{r}_0^T\mathbb{F}_0\boldsymbol{r}_0}\right]\boldsymbol{r}_0$$

** Schwartz inequality

For a pair of vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ in a vector space V, we have by Schwartz inequality:

$$\left|\left(\boldsymbol{u},\boldsymbol{v}\right)\right|\le\left\|\boldsymbol{u}\right\|\left\|\boldsymbol{v}\right\| \tag{i}$$

$\left(\boldsymbol{u},\boldsymbol{v}\right)$ stands for an inner product $\|\cdot\|$ for Euclidean norm.

$$= \left[\boldsymbol{r}_0 + \frac{s_0^2 \boldsymbol{d}_0 \left(\boldsymbol{d}_0^T \boldsymbol{r}_0 \right)}{s_0 \left(\boldsymbol{d}_0^T \boldsymbol{r}_0 \right)} - \frac{\boldsymbol{r}_0 \left(\boldsymbol{r}_0^T \boldsymbol{r}_0 \right)}{\left(\boldsymbol{r}_0^T \boldsymbol{r}_0 \right)} \right]$$

$$= s_0 \boldsymbol{d}_0 = \boldsymbol{p}_0 \quad \left(\text{since the first and third terms cancel out} \right) \tag{2.74a}$$

So, for $k = 0$, Equation (2.72) holds. Similarly, considering Equation (2.71), one has for $k = 0$:

$$\boldsymbol{d}_1^T \boldsymbol{H} \boldsymbol{d}_0 = -\boldsymbol{g}_1^T \mathbb{F}_1 \boldsymbol{H} \boldsymbol{d}_0 \left(\text{since } \boldsymbol{d}_1 = -\mathbb{F}_1 \boldsymbol{g}_1 \text{ and } \mathbb{F}_1 \text{ is symmetric} \right)$$

$$= \frac{-\boldsymbol{g}_1^T \mathbb{F}_1 \boldsymbol{H} \boldsymbol{p}_0}{s_0} \quad \left(\text{since } s_0 \boldsymbol{d}_0 = \boldsymbol{p}_0 \right)$$

$$= -\boldsymbol{g}_1^T \boldsymbol{d}_0 \quad \left(\text{since } \boldsymbol{H} \boldsymbol{p}_0 = \boldsymbol{r}_0, \frac{\mathbb{F}_1 \boldsymbol{r}_0}{s_0} = \boldsymbol{d}_0 \right)$$

$$= 0 \quad \left(\text{from Equation 2.64} \right) \tag{2.74b}$$

Thus (2.71) and (2.72) are satisfied for $k = 0$. Now, assume them to hold for the k^{th} iteration, i.e., $\mathbb{F}_k \boldsymbol{r}_j = \boldsymbol{p}_j, 0 \le j < k$ and $\boldsymbol{d}_k^T \boldsymbol{H} \boldsymbol{d}_j = 0, \; 0 \le j < k$. We proceed to prove that they hold for $k = k+1$ as well. Let us first prove $\mathbb{F}_{k+1} \boldsymbol{r}_j = \boldsymbol{p}_j, 0 \le j < k+1$. From the DFP formula, one has:

$$\mathbb{F}_{k+1} \boldsymbol{r}_j = \left[\mathbb{F}_k + \frac{\boldsymbol{p}_k \boldsymbol{p}_k^T}{\boldsymbol{p}_k^T \boldsymbol{r}_k} - \frac{\mathbb{F}_k \boldsymbol{r}_k \boldsymbol{r}_k^T \mathbb{F}_k}{\boldsymbol{r}_k^T \mathbb{F}_k \boldsymbol{r}_k} \right] \boldsymbol{r}_j, 0 \le j < k+1$$

$$= \left[\mathbb{F}_k \boldsymbol{r}_j + \frac{\boldsymbol{p}_k \boldsymbol{p}_k^T \boldsymbol{r}_j}{\boldsymbol{p}_k^T \boldsymbol{r}_k} - \frac{\mathbb{F}_k \boldsymbol{r}_k \boldsymbol{r}_k^T \mathbb{F}_k \boldsymbol{r}_j}{\boldsymbol{r}_k^T \mathbb{F}_k \boldsymbol{r}_k} \right], 0 \le j < k+1$$

$$= \left[\boldsymbol{p}_j + \frac{s_j \boldsymbol{p}_k \boldsymbol{p}_k^T \boldsymbol{H} \boldsymbol{d}_j}{\boldsymbol{p}_k^T \boldsymbol{r}_k} - \frac{\mathbb{F}_k \boldsymbol{r}_k \boldsymbol{r}_k^T \mathbb{F}_k \boldsymbol{r}_j}{\boldsymbol{r}_k^T \mathbb{F}_k \boldsymbol{r}_k} \right],$$

$$0 \le j < k+1 \left(\text{since } \boldsymbol{r}_j = \boldsymbol{H} \boldsymbol{p}_j = s_j \boldsymbol{H} \boldsymbol{d}_j \right) \tag{2.75}$$

With $\boldsymbol{p}_k^T \boldsymbol{H}\boldsymbol{d}_j = s_k \boldsymbol{d}_k^T \boldsymbol{H}\boldsymbol{d}_j = 0, 0 \le j < k$ because of the $\boldsymbol{H}$-conjugacy of $\boldsymbol{d}_k$, the second term on the RHS of Equation (2.75) is zero. Regarding the third term, the numerator contains $\boldsymbol{r}_k^T \mathbb{F}_k \boldsymbol{r}_j$ which may be written as:

$$\begin{aligned}\boldsymbol{r}_k^T \mathbb{F}_k \boldsymbol{r}_j &= \boldsymbol{r}_k^T \boldsymbol{p}_j \text{ (since } \mathbb{F}_k \boldsymbol{r}_j = \boldsymbol{p}_j, 0 \le j < k \text{ by the induction hypothesis)} \\ &= \boldsymbol{p}_k^T \boldsymbol{H}\boldsymbol{p}_j \text{ (since } \boldsymbol{r}_k = \boldsymbol{H}\boldsymbol{p}_k) \\ &= s_k s_j \boldsymbol{d}_k^T \boldsymbol{H}\boldsymbol{d}_j = 0 \left(\text{by the } \boldsymbol{H}\text{-conjugacy of } \boldsymbol{d}_k\right) \end{aligned} \tag{2.76}$$

The second and third terms on the RHS of Equation (2.75) being thus zero, we get the required result in Equation (2.72), i.e. $\mathbb{F}_{k+1} \boldsymbol{r}_j = \boldsymbol{p}_j, 0 \le j < k+1$. This result helps in proving the $\boldsymbol{H}$-conjugacy of $\boldsymbol{d}_{k+1}$.

Consider the new update at the k^{th} iteration: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + s_k \boldsymbol{d}_k = \boldsymbol{x}_k - s_k \mathbb{F}_k \boldsymbol{g}_k$. The gradient $\nabla f\left(\boldsymbol{x}_{k+1}\right)$ is:

$$\begin{aligned}\nabla f\left(\boldsymbol{x}_{k+1}\right) &= \boldsymbol{g}_{k+1} \\ &= \boldsymbol{g}_{j+1} + \left(\boldsymbol{g}_{j+2} - \boldsymbol{g}_{j+1}\right) + \left(\boldsymbol{g}_{j+3} - \boldsymbol{g}_{j+2}\right) + \ldots + \left(\boldsymbol{g}_{k+1} - \boldsymbol{g}_k\right), 0 \le j \le k \\ &= \boldsymbol{g}_{j+1} + \boldsymbol{r}_{j+1} + \boldsymbol{r}_{j+2} + \ldots + \boldsymbol{r}_k, 0 \le j \le k \end{aligned} \tag{2.77}$$

From Equation (2.73), the last equation takes the form:

$$\boldsymbol{g}_{k+1} = \boldsymbol{g}_{j+1} + \boldsymbol{H}\boldsymbol{p}_{j+1} + \boldsymbol{H}\boldsymbol{p}_{j+2} + \ldots + \boldsymbol{H}\boldsymbol{p}_k, 0 \le j \le k \tag{2.78a}$$

With $\boldsymbol{p}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k = s_k \boldsymbol{d}_k$, one has:

$$\boldsymbol{g}_{k+1} = \boldsymbol{g}_{j+1} + s_{j+1} \boldsymbol{H}\boldsymbol{d}_{j+1} + s_{j+2} \boldsymbol{H}\boldsymbol{d}_{j+2} + \ldots + s_k \boldsymbol{H}\boldsymbol{d}_k, 0 \le j \le k \tag{2.78b}$$

By the induction hypothesis, $\boldsymbol{d}_j^T \boldsymbol{H}\boldsymbol{d}_i = 0$, $i = j+1, j+2, \ldots, k$. Also $\boldsymbol{g}_{j+1}^T \boldsymbol{d}_j = 0$ from Equation (2.64). It follows from Equation (2.78b):

$$\boldsymbol{g}_{k+1}^T \boldsymbol{d}_j = 0, 0 \le j \le k \tag{2.79}$$

Since $\mathbb{F}_{k+1} \boldsymbol{r}_j = \boldsymbol{p}_j, 0 \le j < k+1$, one may write:

$$\boldsymbol{g}_{k+1}^T \mathbb{F}_{k+1} \boldsymbol{r}_j = \boldsymbol{g}_{k+1}^T \boldsymbol{p}_j, 0 \le j < k+1$$

$$\Rightarrow \boldsymbol{g}_{k+1}^T \mathbb{F}_{k+1} \boldsymbol{H}\boldsymbol{p}_j = \boldsymbol{g}_{k+1}^T \boldsymbol{p}_j, 0 \le j < k+1 \left(\text{since } \boldsymbol{r}_j = \boldsymbol{H}\boldsymbol{p}_j \text{ from Equation 2.73}\right) \tag{2.80}$$

With $\boldsymbol{p}_j = \boldsymbol{x}_{j+1} - \boldsymbol{x}_j = s_j \boldsymbol{d}_j$ and $\boldsymbol{d}_j = -\mathbb{F}_j \boldsymbol{g}_j$, the last equation may be written as:

$$\boldsymbol{d}_{k+1}^T \boldsymbol{H} \boldsymbol{d}_j = \boldsymbol{g}_{k+1}^T \boldsymbol{d}_j = 0, 0 \le j < k+1 \ \left(\text{from Equation } 2.79\right) \tag{2.81}$$

♦

Thus $\boldsymbol{d}_{k+1}$ is $\boldsymbol{H}$-conjugate to the search directions of the previous iterations. By proving i) $\boldsymbol{H}$-conjugacy of the search directions $\boldsymbol{d}_k$ and ii) the characteristic of the Hessian inverse $\mathbb{F}_k$ as constructed at the k^{th} iteration by the DFP formula, we realize that the DFP method replicates the CG method and converges in n iterations for a quadratic function. Figure 2.9 shows the convergence of the DFP method as applied to a quadratic and a non-quadratic function.

2.3.2 Broyden-Fletcher-Goldfarb-Shanno (BFGS) Method

Unlike the DFP method, the BFGS method approximates the Hessian matrix itself at each iteration. Let the approximated Hessian matrix be $\mathbb{F}_k$ at the k^{th} iteration. With the update being $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + s_k \boldsymbol{d}_k$, the search direction $\boldsymbol{d}_k$ in this method is given by:

$$\boldsymbol{d}_k = -\boldsymbol{H}_k^{-1} \boldsymbol{g}_k \tag{2.82}$$

The Hessian $\boldsymbol{H}$ of a quadratic function satisfies:

$$\boldsymbol{H}\left[\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\right] = \left[\boldsymbol{r}_0, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_n\right] \tag{2.83}$$

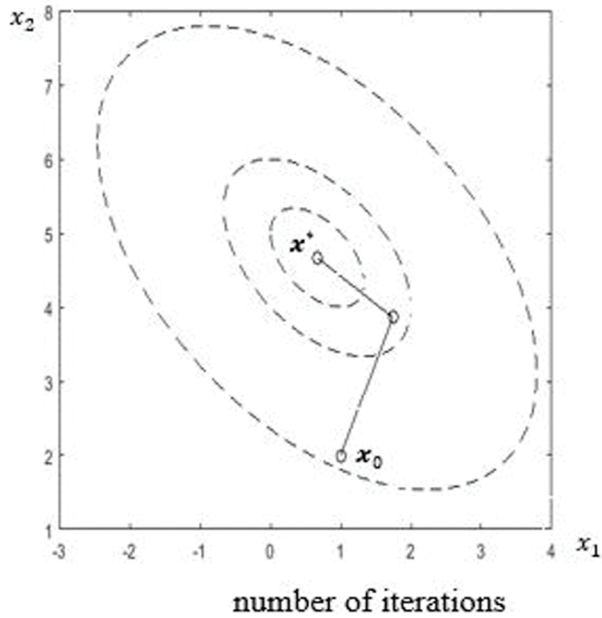
with $\boldsymbol{p}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$ and $\boldsymbol{r}_k = \nabla f\left(\boldsymbol{x}_{k+1}\right) - \nabla f\left(\boldsymbol{x}_k\right) = \boldsymbol{g}_{k+1} - \boldsymbol{g}_k$. It follows that $\boldsymbol{H}_{k+1}$ of the BFGS method satisfies:

$$\boldsymbol{H}_{k+1} \boldsymbol{p}_j = \boldsymbol{r}_j, 0 \le j < k+1 \tag{2.84}$$

This is similar to the property of $\mathbb{F}_k$ in the DFP method. Being an inverse Hessian, $\mathbb{F}_k$ has been shown to satisfy $\mathbb{F}_{k+1} \boldsymbol{r}_j = \boldsymbol{p}_j, 0 \le j < k+1$. Similar to the DFP formula in Equation (2.62), it is possible to estimate $\boldsymbol{H}_{k+1}$ at the k^{th} iteration as a rank two correction to $\boldsymbol{H}_k$. Thus:

$$\boldsymbol{H}_{k+1} = \left[\boldsymbol{H}_k + \frac{\boldsymbol{r}_k \boldsymbol{r}_k^T}{\boldsymbol{r}_k^T \boldsymbol{p}_k} - \frac{\boldsymbol{H}_k \boldsymbol{p}_k \boldsymbol{p}_k^T \boldsymbol{H}_k}{\boldsymbol{p}_k^T \boldsymbol{H}_k \boldsymbol{p}_k}\right] \tag{2.85}$$

In arriving at $\boldsymbol{H}_{k+1}$, we note that $\boldsymbol{p}_k$ and $\boldsymbol{r}_k$ change their positions when compared to the DFP formula in (2.62). This is clear from the relationships that $\mathbb{F}_{k+1}$ and



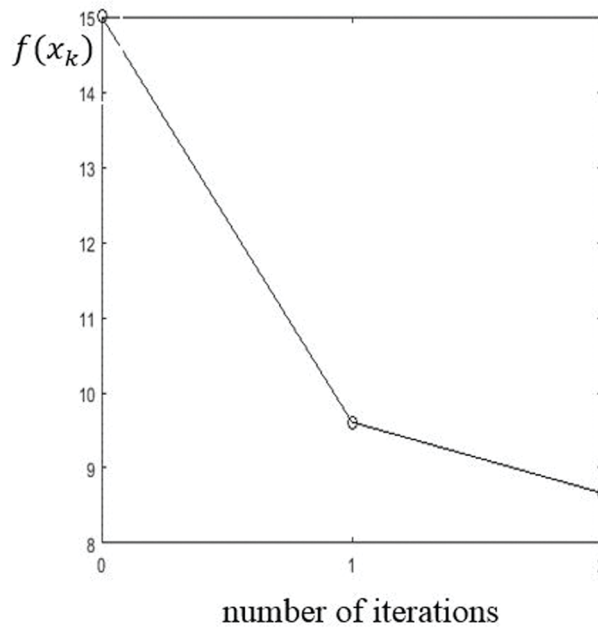


FIGURE 2.9a–b DFP method, quadratic function $f(x_1, x_2) = (x_1 - 3)^2 + (x^2 - 5)^2 + x_1 x_2$; $\boldsymbol{x}_0 = (1,2)^T$; $\boldsymbol{x}^* = (0.667,4.667)^T$ and $f(\boldsymbol{x}^*) = 8.667$ - convergence in two iterations.

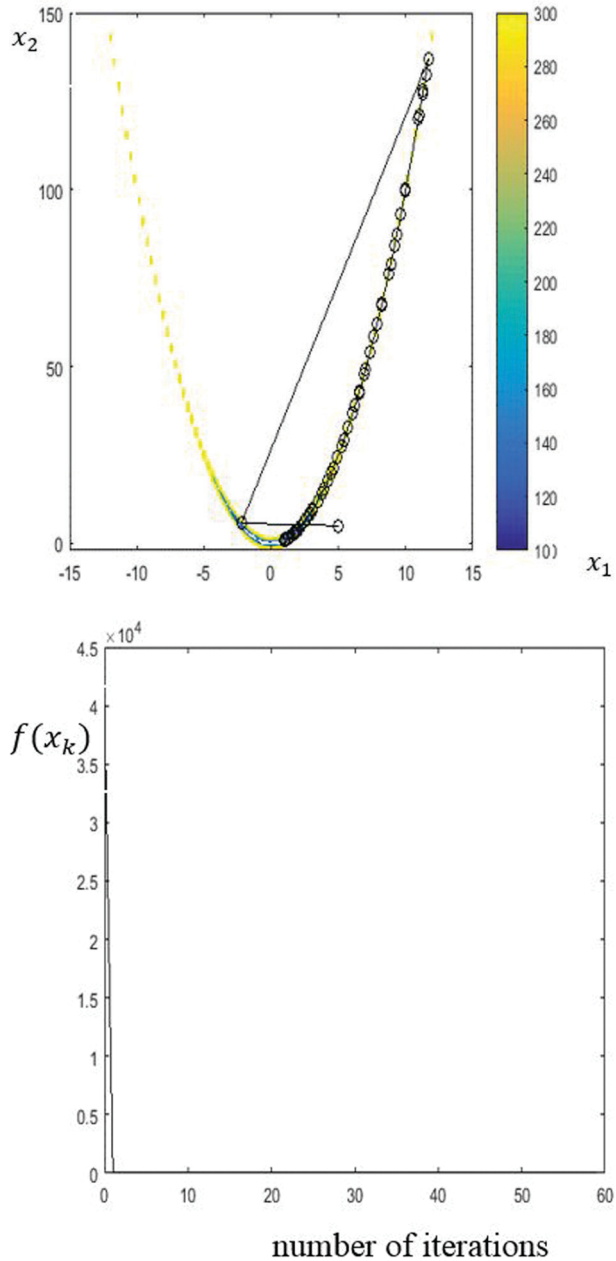


FIGURE 2.9c–d DFP method, non-quadratic function (Rosenbrock)- $f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2$, $\boldsymbol{x}_0 = (5.5)^T$, $\boldsymbol{x}^* = (1,1)^T$ and $f(\boldsymbol{x}^*) = 6.55E - 12$-convergence in 59 iterations.

$\boldsymbol{H}_{k+1}$ have with $\boldsymbol{p}_j$ and $\boldsymbol{r}_j$ (Equations 2.72 and 2.84). Utilizing the symmetry and positive definiteness of $\boldsymbol{H}_k$ and following the method of induction as in the DFP method, one can prove that the search directions in Equation (2.82) are $\boldsymbol{H}$ -conjugate and $\boldsymbol{H}_{k+1}$ satisfies Equation (2.84). At the implementation level, to alleviate the problem of taking an inverse of $\boldsymbol{H}_k$ at each iteration – especially for large-size problems – it is possible to approximate the inverse by using Sherman-Morrison formula[††] (Bartlett 1951).

From Equation (2.85), the inverse is:

$$\boldsymbol{H}_{k+1}^{-1} = \boldsymbol{H}_k^{-1} + \left(1 + \frac{\boldsymbol{r}_k^T \boldsymbol{H}_k^{-1} \boldsymbol{r}_k}{\boldsymbol{p}_k^T \boldsymbol{r}_k}\right) \frac{\boldsymbol{p}_k \boldsymbol{p}_k^T}{\boldsymbol{p}_k^T \boldsymbol{r}_k} - \frac{\boldsymbol{p}_k \boldsymbol{r}_k^T \boldsymbol{H}_k^{-1} + \boldsymbol{H}_k^{-1} \boldsymbol{r}_k \boldsymbol{p}_k^T}{\boldsymbol{r}_k^T \boldsymbol{p}_k} \tag{2.86}$$

In obtaining the inverse in Equation (2.86), the Sherman-Morrison inversion formula is applied twice (Jennifer and Roummel 2017) to $\boldsymbol{H}_{k+1}$ in Equation (2.85).

2.4 PENALTY FUNCTION METHODS

The derivative-based methods discussed so far are meant for unconstrained optimization problems. As is known, problems encountered in practice are almost invariably constrained. With the method of Lagrange multipliers explained in Section 1.6.1, Chapter 1 provides a basic approach to convert the constrained problem into an unconstrained one. As a sequel to this method, we consider penalty function methods (Fiacco and McCormick 1964, 1968) as a computationally efficient and convenient alternative for optimization problems with constraints. Here, using a penalty parameter, we convert a constrained problem into a sequence of unconstrained optimization problems. The interior and exterior variants of the penalty function method are also referred to as sequential unconstrained minimization techniques (SUMTs).

2.4.1 Exterior Penalty Function Method

Let $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ be a scalar objective function, with the equality and inequality constraints given by:

$$h_i(\boldsymbol{x}) = 0, i = 1,2,\ldots,l \text{ and } g_j(\boldsymbol{x}) \leq 0, j = 1,2,\ldots,m \tag{2.87}$$

[††] *Sherman-Morrison formula:* Given a matrix $\boldsymbol{A}$ and vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, the formula for inversion is:

$$\left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^T\right)^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}} \tag{i}$$

A penalty function $\psi(\boldsymbol{x})$ is defined as:

$$\psi(\boldsymbol{x}) = \sum_{i=1}^{l} \left[h_i(\boldsymbol{x})\right]^2 + \sum_{j=1}^{m} \left\{\max\left[0, g_j(\boldsymbol{x})\right]\right\}^2 \tag{2.88}$$

The function $\max\left[0, g_j(\boldsymbol{x})\right]$ outputs a positive non-zero value only when the constraint is violated. Similarly, when the equality constraint is violated (i.e. when $h_i(\boldsymbol{x}) \neq 0$), the penalty function imposes a positive penalty. It imposes no penalty for a feasible point (i.e. when $h_i(\boldsymbol{x}) = 0$). Combining $f(\boldsymbol{x})$ and $\psi(\boldsymbol{x})$ gives the unconstrained optimization problem:

$$\text{minimize } \hat{f}(r, \boldsymbol{x}) = f(\boldsymbol{x}) + r\left(\sum_{i=1}^{l} \left[h_i(\boldsymbol{x})\right]^2 + \sum_{j=1}^{m} \left\{\max\left[0, g_j(\boldsymbol{x})\right]\right\}^2\right) \tag{2.89}$$

$r > 0$ is the penalty parameter, which penalizes a constraint violation. $\hat{f}(r, \boldsymbol{x})$ is the augmented objective function. In the exterior penalty function method, one chooses a positive r and a starting point $\boldsymbol{x}_0$ in the infeasible region. One then solves the resulting unconstrained problem in Equation (2.89). Pending further discussion on the implementation and convergence issues, we note that a selection of proper value for the penalty parameter is indeed tricky. A high value renders the problem highly nonlinear. In solving Equation (2.89) for an unconstrained optimum with an initial choice of r, it is sequentially increased in steps as $r_k = cr_{k-1}$ with $c > 1$. The unconstrained problem corresponding to each r_k is solved by any of the methods presented earlier. As r tends to infinity, the unconstrained minimum so obtained reaches the optimum $\boldsymbol{x}^*$ of the original constrained optimization problem. A proof is provided at the end of the section. From the definition of the penalty function $\psi(\boldsymbol{x})$, it also follows that, for $\boldsymbol{x}$ far away from the feasible region, the penalty increases and the unconstrained minimum is drawn towards the feasible region.

As an illustration, consider the following simple example: minimization of a 1-D problem.

$$\begin{aligned} &\text{minimize } f(x) = (x-4)^2 + 1 \\ &\text{s.t. } x \leq 3 \end{aligned} \tag{2.90}$$

Equation (2.89) gives the unconstrained optimization problem as:

$$\hat{f}(r, x) = (x-4)^2 + 1 + r\left\{\max\left[0, (x-3)\right]\right\}^2 \tag{2.91}$$

Suppose that we start with $r_0 = 10$ and $x_0 = 5$. The unconstrained problems in Equation (2.91) for a sequence of $r_k = 10r_{k-1}$ are solved and Figure 2.10 shows the

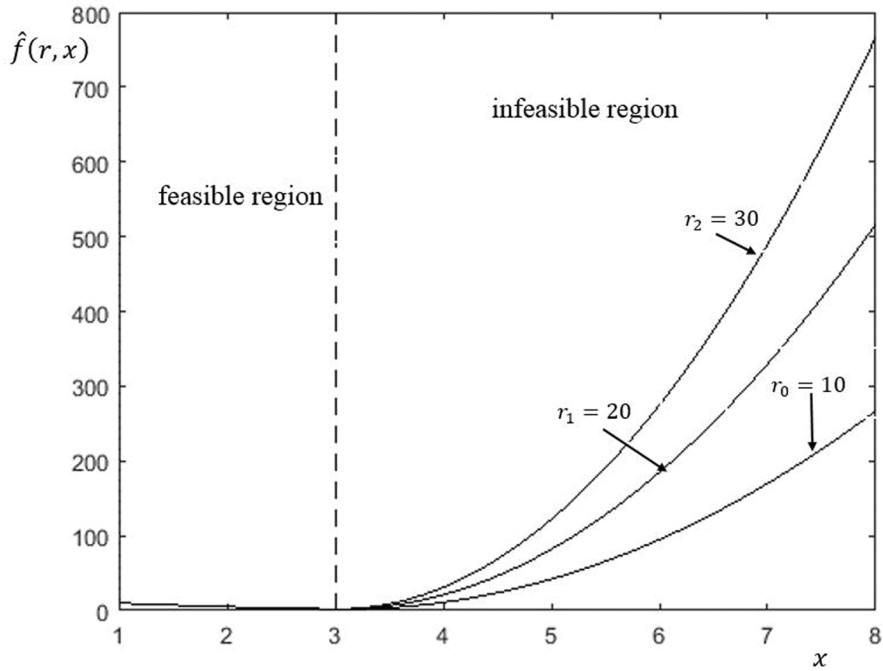


FIGURE 2.10 Exterior penalty function method; unconstrained minima for increasing values of the penalty parameter r with $r_k > r_{k-1} > \ldots > r_0$ tending towards the constrained minimum.

unconstrained minima for $k = 0,1$ and 2 . Obviously, the minimum for the constrained optimization problem in Equation (2.90) is $x^* = 3$, the boundary defined by the constraint as shown in the figure.

Proof for convergence of the penalty function method

Let the penalty method generate $\boldsymbol{x}_k^*$, the sequence of unconstrained minima of $\hat{f}\left(r_k, \boldsymbol{x}_k^*\right)$ with respect to r_k, $k = 0,1,\ldots$. We show that $\hat{f}\left(r_k, \boldsymbol{x}_k^*\right)$ is a monotonic and convergent sequence. Further, the limit of any convergent subsequence of $\left\{\boldsymbol{x}_k^*\right\}$ is an optimal solution. The proof closely follows the one in Bazaraa *et al.* (2006). Before it is presented, we first prove the following inequalities:

$$\text{(i)} \quad \hat{f}\left(r_k, \boldsymbol{x}_k^*\right) \le \hat{f}\left(r_{k+1}, \boldsymbol{x}_{k+1}^*\right)$$

$$\text{(ii)} \quad \psi\left(\boldsymbol{x}_k^*\right) \ge \psi\left(\boldsymbol{x}_{k+1}^*\right)$$

$$\text{(iii)} \quad f\left(\boldsymbol{x}_k^*\right) \le f\left(\boldsymbol{x}_{k+1}^*\right) \qquad \text{(2.92a,b,c)}$$

Considering the RHS of (2.92a), we have:

$$\hat{f}\left(r_{k+1}, \boldsymbol{x}_{k+1}^*\right) = f\left(\boldsymbol{x}_{k+1}^*\right) + r_{k+1}\psi\left(\boldsymbol{x}_{k+1}^*\right)$$

$$\geq f\left(\boldsymbol{x}_{k+1}^*\right) + r_k \psi\left(\boldsymbol{x}_{k+1}^*\right) \qquad \left(\text{since } r_k < r_{k+1}\right)$$

$$\geq f\left(\boldsymbol{x}_k^*\right) + r_k \psi\left(\boldsymbol{x}_k^*\right) = \hat{f}\left(r_k, \boldsymbol{x}_k^*\right)$$

$$\left(\text{since } \boldsymbol{x}_k^* \text{ is the unconstrained minimum for } r = r_k\right) \tag{2.93}$$

which proves the first inequality (2.92a). Now consider the following two inequalities.

$$\hat{f}\left(r_k, \boldsymbol{x}_k^*\right) = f\left(\boldsymbol{x}_k^*\right) + r_k \psi\left(\boldsymbol{x}_k^*\right) \leq f\left(\boldsymbol{x}_{k+1}^*\right) + r_k \psi\left(\boldsymbol{x}_{k+1}^*\right) \tag{2.94}$$

and

$$\hat{f}\left(r_{k+1}, \boldsymbol{x}_{k+1}^*\right) = f\left(\boldsymbol{x}_{k+1}^*\right) + r_{k+1}\psi\left(\boldsymbol{x}_{k+1}^*\right) \leq f\left(\boldsymbol{x}_k^*\right) + r_{k+1}\psi\left(\boldsymbol{x}_k^*\right) \tag{2.95}$$

Adding the last two inequalities, one has:

$$f\left(\boldsymbol{x}_k^*\right) + r_k \psi\left(\boldsymbol{x}_k^*\right) + f\left(\boldsymbol{x}_{k+1}^*\right) + r_{k+1}\psi\left(\boldsymbol{x}_{k+1}^*\right) \leq f\left(\boldsymbol{x}_{k+1}^*\right) + r_k \psi\left(\boldsymbol{x}_{k+1}^*\right)$$

$$+ f\left(\boldsymbol{x}_k^*\right) + r_{k+1}\psi\left(\boldsymbol{x}_k^*\right)$$

$$\Rightarrow r_k \psi\left(\boldsymbol{x}_k^*\right) + r_{k+1}\psi\left(\boldsymbol{x}_{k+1}^*\right) \leq r_k \psi\left(\boldsymbol{x}_{k+1}^*\right) + r_{k+1}\psi\left(\boldsymbol{x}_k^*\right)$$

$$\Rightarrow \left(r_{k+1} - r_k\right)\psi\left(\boldsymbol{x}_{k+1}^*\right) \leq \left(r_{k+1} - r_k\right)\psi\left(\boldsymbol{x}_k^*\right)$$

$$\Rightarrow \psi\left(\boldsymbol{x}_{k+1}^*\right) \leq \psi\left(\boldsymbol{x}_k^*\right)\left(\text{since } r_{k+1} - r_k > 0\right) \tag{2.96}$$

The last step above proves the second inequality (2.92b). From the last step in arriving at (2.93), one has:

$$f\left(\boldsymbol{x}_{k+1}^*\right) + r_k \psi\left(\boldsymbol{x}_{k+1}^*\right) \geq f\left(\boldsymbol{x}_k^*\right) + r_k \psi\left(\boldsymbol{x}_k^*\right)$$

$$\Rightarrow f\left(\boldsymbol{x}_{k+1}^*\right) \geq f\left(\boldsymbol{x}_k^*\right) + r_k\left(\psi\left(\boldsymbol{x}_k^*\right) - \psi\left(\boldsymbol{x}_{k+1}^*\right)\right)$$

$$\Rightarrow f\left(\boldsymbol{x}_{k+1}^*\right) \geq f\left(\boldsymbol{x}_k^*\right)\left(\text{from } 2.96\right) \tag{2.97}$$

This proves the third inequality (2.92c). To arrive the result on convergence of the method, we go through the following steps.

Step 1. Consider a feasible point $\boldsymbol{x}$ such that $\psi(\boldsymbol{x}) = 0$, i.e., $g(\boldsymbol{x}) \le 0$ and $h(\boldsymbol{x}) = 0$. One has for each r_k:

$$f(\boldsymbol{x}) = f(\boldsymbol{x}) + r_k \psi(\boldsymbol{x}) \ge f(\boldsymbol{x}_k^*) + r_k \psi(\boldsymbol{x}_k^*) = \hat{f}(r_k, \boldsymbol{x}_k^*) \tag{2.98}$$

Hence, if $\boldsymbol{x}^*$ is the optimal solution to the constrained problem, then:

$$\inf\{f(\boldsymbol{x}) : \boldsymbol{x} \text{ feasible}\} = f(\boldsymbol{x}^*) \ge \sup_{r_k \ge 0} \hat{f}(r_k, \boldsymbol{x}_k^*) = \lim_{r_k \to \infty} \hat{f}(r_k, \boldsymbol{x}_k^*) \tag{2.99}$$

The equality in the last statement above is due to (2.92a) where it is shown that $\hat{f}(r_k, \boldsymbol{x}_k^*)$ is a monotone (a non-decreasing sequence).

Step 2. Next we show that $\psi(\boldsymbol{x}_k^*) \to 0$ as $k \to \infty$, i.e. $r_k \to \infty$. Towards this, let $\boldsymbol{y}$ be a feasible point and $\boldsymbol{x}_L^*$ an optimal solution with $r_L = 1$ so that $f(\boldsymbol{x}_L^*) = \inf f(\boldsymbol{y} + \psi(\boldsymbol{y}))$. Assuming r_k to be:

$$r_k \ge \left(\frac{1}{\varepsilon}\right) \left| f(\boldsymbol{y}) - f(\boldsymbol{x}_L^*) \right| + 2, \varepsilon > 0 \tag{2.100}$$

the inequality (2.97) gives:

$$f(\boldsymbol{x}_k^*) \ge f(\boldsymbol{x}_L^*) \tag{2.101}$$

Hence,

$$\hat{f}(r_k, \boldsymbol{x}_k^*) = f(\boldsymbol{x}_k^*) + r_k \psi(\boldsymbol{x}_k^*) \ge f(\boldsymbol{x}_L^*) + r_k \psi(\boldsymbol{x}_k^*) \tag{2.102}$$

If $\psi(\boldsymbol{x}_k^*) > \varepsilon$, then from the definition of r_k, the inequality in (2.102) takes the form:

$$\hat{f}(r_k, \boldsymbol{x}_k^*) \ge f(\boldsymbol{x}_L^*) + \left| f(\boldsymbol{y}) - f(\boldsymbol{x}_L^*) \right| + 2\varepsilon \ge f(\boldsymbol{y}) \tag{2.103}$$

In view of (2.98), the above is not valid since $\boldsymbol{y}$ is a feasible point. So, by contradiction with the earlier supposition that $\psi(\boldsymbol{x}_k^*) > \varepsilon$, we get:

$$\psi\left(\boldsymbol{x}_k^*\right) \leq \varepsilon \tag{2.104a}$$

Since $\varepsilon(>0)$ is arbitrary,

$$\lim_{k\to\infty} \psi\left(\boldsymbol{x}_k^*\right) \to 0 \tag{2.104b}$$

Step 3. Now, consider a subsequence $\left\{\boldsymbol{y}_k^*\right\}$ of $\left\{\boldsymbol{x}_k^*\right\}$ and let $\bar{\boldsymbol{y}}$ be its limit. We need to show that $\bar{\boldsymbol{y}}$ is indeed an optimal solution to the original problem.

$$\hat{f}\left(r_k, \boldsymbol{y}_k^*\right) = f\left(\boldsymbol{y}_k^*\right) + r_k \psi\left(\boldsymbol{y}_k^*\right) \geq f\left(\boldsymbol{y}_k^*\right) \tag{2.105}$$

Since $\boldsymbol{y}_k^* \to \bar{\boldsymbol{y}}$ and $f(\boldsymbol{x})$ is continuous,

$$\sup_{r_k \geq 0} \hat{f}\left(r_k, \boldsymbol{y}_k^*\right) = \lim_{r_k \to \infty} \hat{f}\left(r_k, \boldsymbol{y}_k^*\right) \geq f\left(\bar{\boldsymbol{y}}\right) \tag{2.106}$$

But $\lim_{k\to\infty} \psi\left(\boldsymbol{y}_k^*\right) \to 0$ from step 2 and this gives $\psi\left(\bar{\boldsymbol{y}}\right) = 0$. Therefore $\bar{\boldsymbol{y}}$ is a feasible point. Thus, combining the assertions in (2.99) of step 1 and (2.106), we get the result that $\bar{\boldsymbol{y}}$ is an optimal solution such that $\sup_{r_k \geq 0} \hat{f}\left(r_k, \boldsymbol{x}_k^*\right) = f\left(\bar{\boldsymbol{y}}\right)$. This completes the proof. ♦

2.4.2 Interior Penalty Function Method

This penalty function method applies to optimization problems with inequality constraints. As the name indicates, the method iterates through the interior (feasible) region of the design space. The method employs the penalty function $\psi(\boldsymbol{x})$ in the following forms:

$$\psi(\boldsymbol{x}) = -\sum_{j=1}^{m} \frac{1}{g_j(\boldsymbol{x})} \text{ or } \psi(\boldsymbol{x}) = -\sum_{j=1}^{m} \log\left(-g_j(\boldsymbol{x})\right) \tag{2.107a,b}$$

and the augmented objective function is:

$$\hat{f}(r, \boldsymbol{x}) = f(\boldsymbol{x}) + r\psi(\boldsymbol{x}) \tag{2.107c}$$

In this method, $r\,(>0)$ corresponds to a decreasing sequence, say, $r_k = cr_{k-1}$ with $c \in (0,1)$. As $r_k \to 0$, $\boldsymbol{x}_k^* \to \boldsymbol{x}^*$ where $\boldsymbol{x}_k^*$ is the unconstrained minimum at the k^{th} iteration. The proof is similar to the one given for the exterior penalty

function method. The constraint functions $g_j(\boldsymbol{x}), j = 1,2,\ldots,m$ are negative in the feasible region. $\hat{f}(r,\boldsymbol{x})$ tends to infinity as $\boldsymbol{x}$ approaches the boundary defined by $\{\boldsymbol{x} : g_j(\boldsymbol{x}) = 0\}$. Thus, in the minimization process of the augmented function $\hat{f}(\boldsymbol{r},\boldsymbol{x})$ for each r_k, $\boldsymbol{x}_k^*$ converges to $\boldsymbol{x}^*$ from within the interior of the feasible region $\{\boldsymbol{x} : g_j(\boldsymbol{x}) \le 0\}$.

Referring to the one-dimensional example in Equation (2.90), if $\psi(x) = -\dfrac{1}{g(x)}$, the augmented function $\hat{f}(r,x)$ is:

$$\hat{f}(r,x) = (x-4)^2 + 1 - r\frac{1}{(x-3)} \tag{2.108}$$

Figure 2.11 shows the convergence to $x^* = 3$ with a decreasing sequence $r_k = 0.5r_{k-1}$ and with $x_0 = 1$.

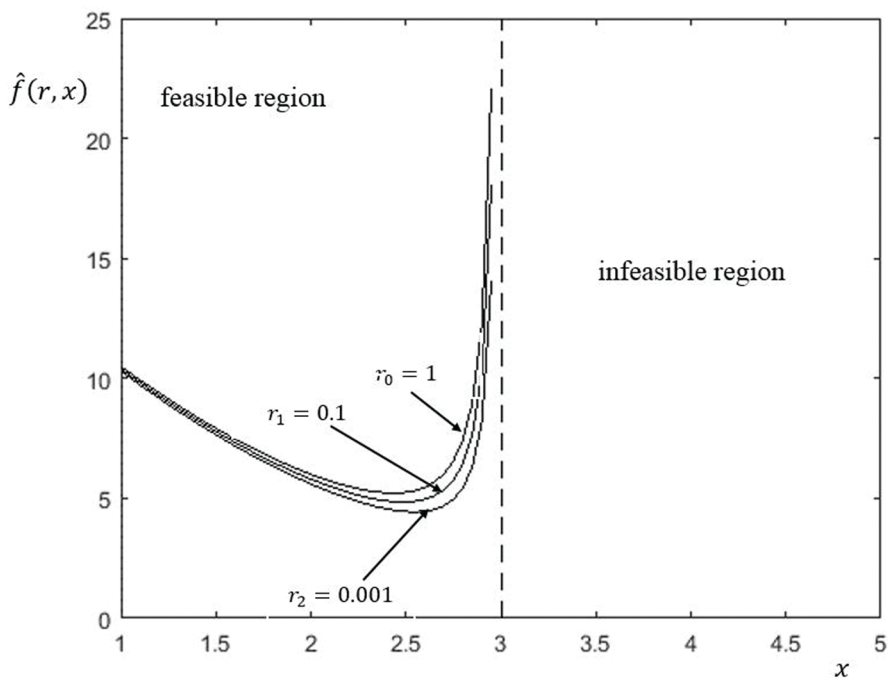


FIGURE 2.11 Interior penalty function method; unconstrained minima for decreasing values of the penalty parameter $r_k < r_{k-1} < \ldots < r_0$ tend to the constrained minimum x^* at the barrier.

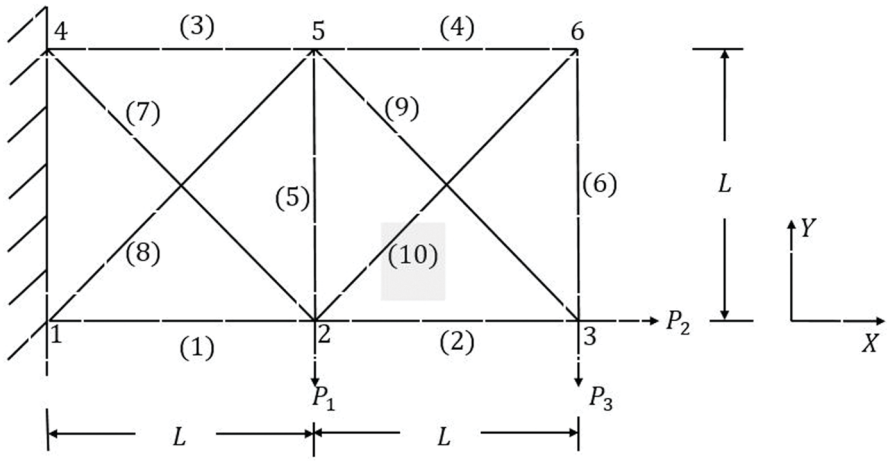


FIGURE 2.12 A 10-member plane truss; FE model with 2 *dof* /node in the two transverse directions, $L = 150\,cm$, mass density = 2700E-6 Kg/cm^3, Young's modulus of elasticity $E = 70 \times 10^5 N/cm^2$, $P_1 = 500\,KN$, $P_2 = 100\,KN$ and $P_3 = 100\,KN$.

Example 2.2. Consider weight minimization of a plane truss (Figure 2.12) under a displacement constraint by the penalty methods. The 10-member truss is subject to static loads as shown. The member areas constitute the design variables with specified lower and upper bounds.

Solution. With the number of design variables denoted by $N = 10$, the constrained optimization problem is:

$$\text{minimize the weight, } W = \rho L\left(\sum_{i=1}^{6} A_i + \sqrt{2}\sum_{i=7}^{N} A_i\right)$$

$$\text{s.t. } 6\,\text{cm}^2 \le A_i, i = 1,2,\ldots,N \le 10\,\text{cm}^2$$

$$\text{and } Y\text{- displacement at node } 3 \le 6 \text{ cm} \tag{2.109}$$

Let $\boldsymbol{x}$ represent the unknown $A_i, i = 1,2,\ldots,N$ (member cross sectional areas in cm^2) which are the design variables. Here, the objective function is $f(\boldsymbol{x}) = W$. Let $\boldsymbol{U} = \left(U_1, U_2, U_3, U_4, U_5, U_6, U_7, U_8\right)^T$ represent the vector of nodal displacements where the pairs $\left(U_1, U_2\right)$, $\left(U_3, U_4\right)$, $\left(U_5, U_6\right)$ and $\left(U_7, U_8\right)$ are displacements in X- and Y-directions, respectively, at nodes 2, 3, 5 and 6 in Figure 2.12. Note that the nodal displacements are implicitly related to the member areas. FEM is used to solve for these displacements (Section 1.5.1, Chapter 1) via the discretized equilibrium equations:

$$[\boldsymbol{K}]\{\boldsymbol{U}\} = \{\boldsymbol{F}\} \tag{2.110}$$

$\boldsymbol{K} \in \mathbb{R}^{n\times n}$ is the stiffness matrix obtained by the assembly of element stiffness matrices where $n = 8$ is the number of *dof*s of the plane truss. The elements of each element matrix $\boldsymbol{K}^e$ are functions of the design variables $\boldsymbol{x}$ and are given in Appendix B. $\boldsymbol{F} \in \mathbb{R}^n$ is the force vector $\{0, P_1, P_2, P_3, 0, 0, 0, 0, 0\}^T$. The CG method is used for unconstrained minimization of the augmented objective function $\hat{f}(r, \boldsymbol{x})$:

$$\hat{f}(r, \boldsymbol{x}) = f(\boldsymbol{x}) + r\psi(\boldsymbol{x}) \tag{2.111}$$

where $\psi(\boldsymbol{x})$ is defined in the interior penalty function method as:

$$\psi(\boldsymbol{x}) = -\sum_{i=1}^{10} \frac{1}{6 - x_i} - \sum_{i=1}^{N} \frac{1}{x_i - 10} - \frac{1}{U_4 - 6} \tag{2.112}$$

The CG method requires the gradient vector $\nabla\hat{f}$ with respect to $\boldsymbol{x}$ to solve the unconstrained problem for each $r_k, k = 1, 2, \ldots$. This vector is given by:

$$\nabla\hat{f} = \nabla f + r\nabla\psi \tag{2.113}$$

where

$$\nabla f = \rho L\left(1, 1, 1, 1, 1, \sqrt{2}, \sqrt{2}, \sqrt{2}, \sqrt{2},\right)^T \tag{2.114}$$

and

$$\nabla\psi_i = -\frac{1}{\left(6 - x_i\right)^2} + \frac{1}{\left(x_i - 10\right)^2} + \frac{1}{\left(U_4 - 6\right)^2}\frac{dU_4}{dx_i} \tag{2.115}$$

To obtain $\dfrac{dU_4}{dx_i}$, we utilize Equation (2.110). Differentiating the equation with respect to $x_i \left(= A_i\right)$ gives:

$$\begin{aligned} &\frac{\partial[\boldsymbol{K}]}{\partial x_i}\{\boldsymbol{U}\} + \boldsymbol{K}\frac{d\boldsymbol{U}}{dx_i} = \{0\}, i = 1, 2, \ldots, N \\ \Rightarrow \frac{d\boldsymbol{U}}{dx_i} &= \left(\frac{dU_1}{dx_i}, \frac{dU_2}{dx_i}, \frac{dU_3}{dx_i}, \frac{dU_4}{dx_i}, \frac{dU_5}{dx_i}, \frac{dU_6}{dx_i}, \frac{dU_7}{dx_i}, \frac{dU_8}{dx_i}\right)^T \\ &= -\boldsymbol{K}^{-1}\frac{\partial[\boldsymbol{K}]}{\partial x_i}\{\boldsymbol{U}\} \end{aligned} \tag{2.116}$$

$\dfrac{\partial[\boldsymbol{K}]}{\partial x_i} \in \mathbb{R}^{n\times n}$ may be called the sensitivity matrix. It is given in Appendix B.

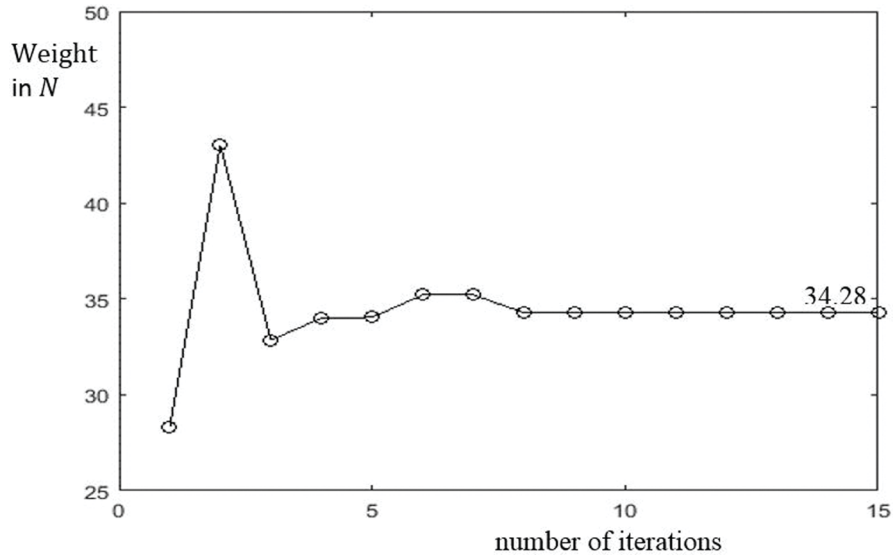


FIGURE 2.13 Weight optimization of a plane truss by interior penalty function method; $r_0 = 1.0$ and $r_k = 0.5 r_{k-1}$, $x_0(1:N) = 6$ sq.cm., $\boldsymbol{x}^* = (9.68, 6.0, 9.65, 9.32, 6.0, 9.32, 6.19, 6.0, 6.13, 6.20)^T$ and at the end of iterations, Y-displacement at node 3 = −5.82 cm.

With the available $\boldsymbol{x}$ at each iteration, $\boldsymbol{K}$ and $\dfrac{\partial[\boldsymbol{K}]}{\partial x_i}$ are computed. $\boldsymbol{U}$ is solved from Equation (2.110). Substitution in (2.116) yields $\dfrac{d\boldsymbol{U}}{dx_i}$ from which $\dfrac{dU_4}{dx_i}$ is utilized in Equation (2.115) to compute the gradient $\nabla \hat{f}$.

With $r_0 = 1000$ as the initial choice, it is sequentially reduced as $r_k = 0.1 r_{k-1}$. The initial vector $\boldsymbol{x}_0(1:N) = (6, 6, \ldots, 6)^T$ in sq.cm. The evolution of the objective function $f(\boldsymbol{x}) = W$, i.e. the weight of the truss, is shown in Figure 2.13.

Result obtained by the exterior penalty function method with the augmented objective function formulated as per Equation (2.89) is shown in Figure 2.14. Here also the unconstrained optimization problem at each iteration is solved by CG method. The search direction $\boldsymbol{d}_k$ is obtained by numerically differentiating of the augmented objective function as per the procedure outlined in Section 2.4.4.

■

Limitations of the penalty function methods

In both the penalty functions methods, the convergence to $\boldsymbol{x}^*$ is guaranteed when the unconstrained minimum is fairly close to $\boldsymbol{x}_k^*$ at each r_k. This is implicit in the proof for convergence. While the methods are easy to adapt computationally and robust to

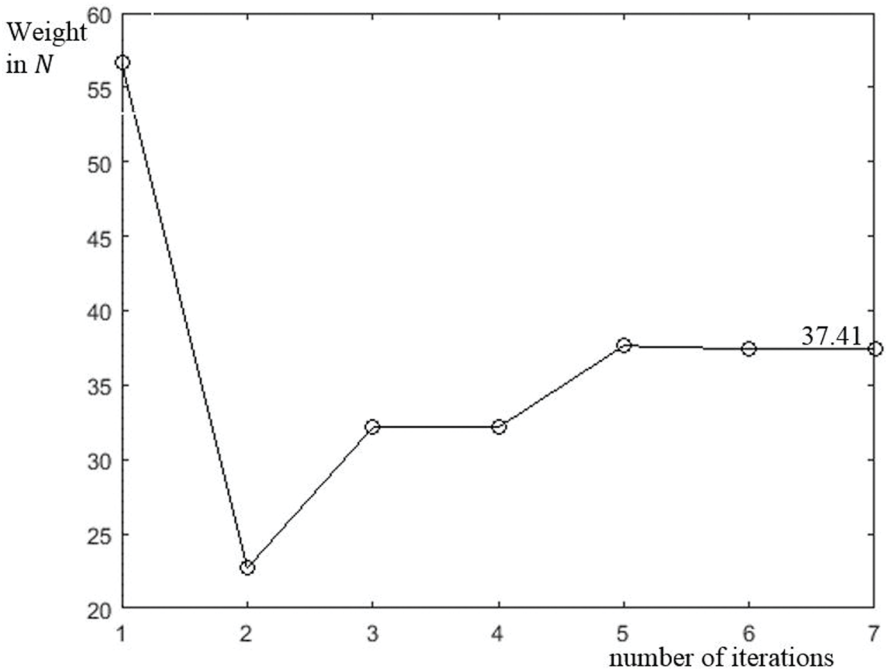


FIGURE 2.14 Plane truss: weight optimization by exterior penalty function method; $r_0 = 10$ and $r_k = 5r_{k-1}$, $x_0(1:N) = 12.0$ sq.cm., $\boldsymbol{x}^* = (8.93, 7.76, 9.53, 7.81, 7.76, 7.81, 6.0, 6.81, 8.66, 8.76)^T$ and at the end of iterations Y-displacement at node 3 = −5.63 cm.

handle constrained optimization problems, even those of large sizes, they do have certain shortcomings. For example, choosing a starting point $\boldsymbol{x}_0$ in higher dimensional design space – especially a feasible point for the interior penalty function method – is far from trivial. Also, as mentioned earlier, choosing the initial value for the penalty parameter r is also difficult for both the methods. The value may be selected as the ratio of the objective function to the penalty function. It may be preceded by a suitable scaling of the constraint functions $g_j(\boldsymbol{x}), j = 1, 2, \ldots, m$ so that the magnitude of the penalty function $\psi(\boldsymbol{x})$ is comparable to that of $f(\boldsymbol{x})$. This ensures that both the objective function and the constraints effectively participate in subsequent changes during the iterations. Note that with increasing r_k, the Hessian of the augmented objective function $\hat{f}(r, \boldsymbol{x})$ may be ill-conditioned affecting the convergence of the method. It is found that by combining the penalty term with Lagrange multipliers, ill-conditioning effects may be avoided. This has led to the development of the augmented Lagrangian method (Hestenes 1969, Powell 1969).

2.4.3 Augmented Lagrangian Method (ALM)

With equality constraints $h_i(\boldsymbol{x}), i = 1, 2, \ldots, l$, the augmented objective function is defined as:

$$\hat{f}\left(\mu, r_k, \boldsymbol{x}\right) = f\left(\boldsymbol{x}\right) + \sum_{i=1}^{l} \mu_i h_i\left(\boldsymbol{x}\right) + r_k \sum_{i=1}^{l} h^2\left(\boldsymbol{x}\right) \tag{2.117}$$

$\mu_i, i = 1,2,\ldots,l$ are the Lagrange multipliers (see the method of Lagrange multipliers, Section 1.6.1, Chapter 1). The KKT condition for optimality gives:

$$\nabla f\left(\boldsymbol{x}_k^*\right) + \sum_{i=1}^{l} \mu_i \nabla h_i\left(\boldsymbol{x}_k^*\right) + 2r_k \sum_{i=1}^{l} h_i\left(\boldsymbol{x}_k^*\right) \nabla h_i\left(\boldsymbol{x}_k^*\right) = 0$$

$$\Rightarrow \nabla f\left(\boldsymbol{x}_k^*\right) + \sum_{i=1}^{l} \left(\mu_i + 2r_k h_i\left(\boldsymbol{x}_k^*\right)\right) \nabla h_i\left(\boldsymbol{x}_k^*\right) = 0 \tag{2.118}$$

The last equation leads to an update formula for the Lagrange multipliers at each iteration as:

$$\mu_{i,k+1} = \mu_{i,k} + 2r_k h_i\left(\boldsymbol{x}_k^*\right) \tag{2.119}$$

Convergence of the method is proved in (Rockafellar 1974, Conn *et al.* 1991). In case of inequality constraints $g_j\left(\boldsymbol{x}\right), j = 1,2,\ldots,m$, the augmented objective function takes the form:

$$\hat{f}\left(\lambda, r_k, \boldsymbol{x}, \boldsymbol{y}\right) = f\left(\boldsymbol{x}\right) + \sum_{j=1}^{m} \lambda_j \left(g_j\left(\boldsymbol{x}\right) + y_j^2\right) + r_k \sum_{j=1}^{m} \left(g_j\left(\boldsymbol{x}\right) + y_j^2\right)^2 \tag{2.120}$$

$\boldsymbol{y} = \left\{y_j, j = 1,2,\ldots,m\right\}$ is the vector of slack variables and $\boldsymbol{\lambda}$ the vector of Lagrange multipliers for the inequality constraints. Here the number of design variables increases to $n + m$. However, the slack variables may be eliminated by the following procedure (Rockafellar 1974). The gradient vector of $\hat{f}\left(\boldsymbol{\lambda}, r_k, \boldsymbol{x}, \boldsymbol{y}\right)$ with respect to the slack variables is:

$$\frac{\partial \hat{f}\left(\boldsymbol{\lambda}, r_k, \boldsymbol{x}, \boldsymbol{y}\right)}{\partial y_j} = 2\lambda_j y_j + 4r_k y_j \left(g_j\left(\boldsymbol{x}\right) + y_j^2\right) \tag{2.121}$$

The optimality condition for the local minimum of $\hat{f}\left(\boldsymbol{\mu}, r_k, \boldsymbol{x}, \boldsymbol{y}\right)$ with respect to $\boldsymbol{y}$ is:

$$\left(\lambda_j y_j + 2r_k y_j \left(g_j\left(\boldsymbol{x}\right) + y_j^2\right)\right) = 0, j = 1,2,\ldots,m \tag{2.122}$$

i.e. either $y_j = 0$

$$\text{or } y_j^2 = -\frac{\lambda_j}{2r_k} - g_j(\boldsymbol{x}) \tag{2.123}$$

Disregarding a negative value for y_j^2, we get a solution for the slack variables as:

$$y_j^2 = \max\left(0, -\frac{\lambda_j}{2r_k} - g_j(\boldsymbol{x})\right)$$

$$\Rightarrow g_j(\boldsymbol{x}) + y_j^2 = \max\left(g_j(\boldsymbol{x}), -\frac{\lambda_j}{2r_k}\right) \tag{2.124}$$

Thus the unconstrained optimization problem for each r_k (Equation 2.120) takes the form:

$$\hat{f}(\boldsymbol{\lambda}, r_k, \boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j \left(\max\left(g_j(\boldsymbol{x}), -\frac{\lambda_j}{2r_k}\right)\right) + r_k \sum_{j=1}^{m} \left(\max\left(g_j(\boldsymbol{x}), -\frac{\lambda_j}{2r_k}\right)\right)^2 \tag{2.125}$$

Similar to the update formula in Equation (2.119), one obtains an update for λ_j as:

$$\lambda_{j,k+1} = \lambda_{j,k} + 2r_k \max\left(g_j(\boldsymbol{x}), -\frac{\lambda_{j,k}}{2r_k}\right) \tag{2.126}$$

Thus, when both equality and inequality constraints apply, we combine Equations (2.117) and (2.125) to get the augmented objective function as:

$$\hat{f}(\boldsymbol{\mu}, \boldsymbol{\lambda}, r_k, \boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i=1}^{l} \mu_i h_i(\boldsymbol{x}) + r_k \sum_{i=1}^{l} h_i^2(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j \left(\max\left(g_j(\boldsymbol{x}), -\frac{\lambda_j}{2r_k}\right)\right)$$

$$+ r_k \sum_{j=1}^{m} \left(\max\left(g_j(\boldsymbol{x}), -\frac{\lambda_j}{2r_k}\right)\right)^2 \tag{2.127}$$

With an initial choice of $\boldsymbol{\mu}$, $\boldsymbol{\lambda}$ and r, ALM obtains the solution to the original problem starting from an infeasible region similar to the exterior penalty function method. Note that the iterative updates in Equations (2.119) and (2.126) help in giving the optimal Lagrange multipliers $\boldsymbol{\mu}^*$ and $\boldsymbol{\lambda}^*$ at the end of the iterative process.

Example 2.3. Consider the constrained optimization of the Rosen-Suzuki function (Vanderplaats 1973):

$$\text{minimize } f(\boldsymbol{x}) = x^2(1) - 5x(1) + x^2(2) - 5x(2) + 2x^2(3) - 21x(3) + x^2(4) + 7x(4) + 50 \tag{2.128a}$$

$$\text{s.t. } h_1(\boldsymbol{x}) = x^2(1) + x(1) + x^2(2) - x(2) + x^2(3) + x(3) + x^2(4) - x(4) - 8 = 0$$

$$h_2(\boldsymbol{x}) = 2x^2(1) + 2x(1) + x^2(2) - x(2) + x^2(3) - x(4) - 5 = 0$$

$$g(\boldsymbol{x}) = x^2(1) - x(1) + 2x^2(2) + x^2(3) + 2x^2(4) - x(4) - 10 \le 0 \tag{2.128b,c,d}$$

Solution. Let μ_1 and μ_2 be the Lagrange multipliers associated with the two equality constraints $h_1(\boldsymbol{x})$ and $h_2(\boldsymbol{x})$ respectively. Let λ be the corresponding multiplier for the inequality constraint $g(\boldsymbol{x})$. The augmented objective function (Equation 2.127) is:

$$\hat{f}(\mu_1, \mu_2, \lambda, r_k, \boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i=1}^{2} \mu_i h_i(\boldsymbol{x}) + r_k \sum_{i=1}^{2} h_i^2(\boldsymbol{x}) + \lambda \max\left(g(\boldsymbol{x}), -\frac{\lambda}{2r_k}\right) + r_k \left(\max\left(g(\boldsymbol{x}), -\frac{\lambda}{2r_k}\right)\right)^2 \tag{2.129}$$

The iterative process is started with $\boldsymbol{x}_0 = (1,1,1,1)^T$, $r_0 = 1$ and $(\mu_1, \mu_2, \lambda) = (0,0,0)$. The unconstrained optimization is performed using the CG method. The constrained optimization results are given in Figures 2.15 and 2.16 which match with the known exact solution (Vanderplaats 1973). The multipliers μ_1 and μ_2 are non-zero (see Figure 2.17a) during the optimization process, even as λ is zero (see Figure 2.17b).

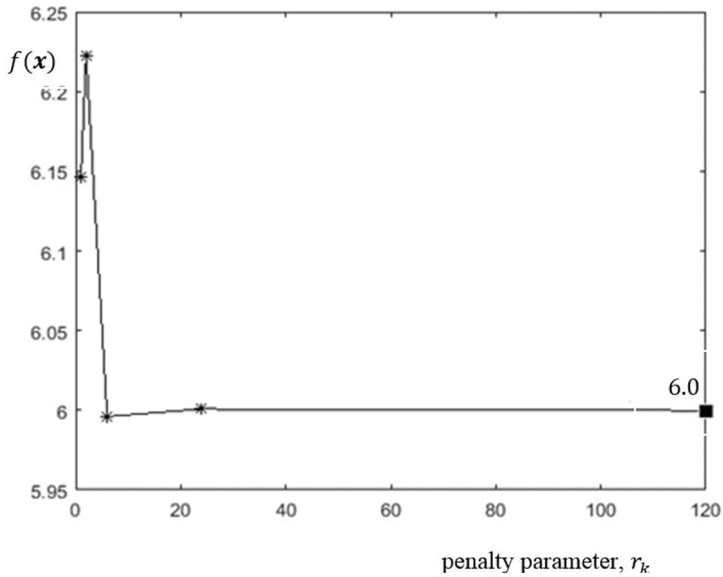


FIGURE 2.15 Augmented Lagrangian method along with CG method; Rosen-Suzuki function (Vanderplaats 1973), evolution of $f(\boldsymbol{x})$ with respect to $r_k = ir_{k-1}$, $i = 1,2,3,4,5$, optimum value $f(\boldsymbol{x}^*) \cong 6.0$.

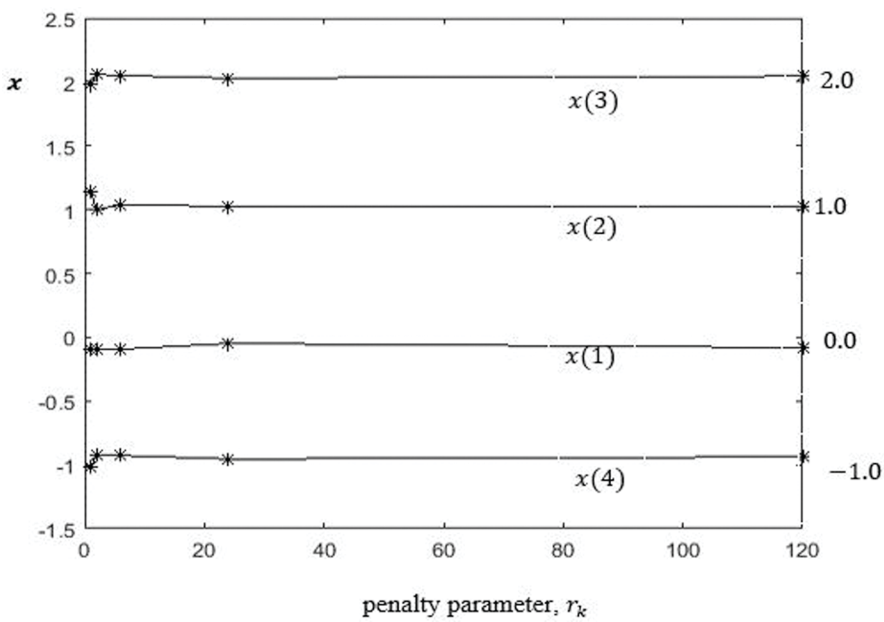


FIGURE 2.16 Augmented Lagrangian method along with CG method; Rosen-Suzuki function (Vanderplaats 1973), evolutions of design variables with respect to $r_k = ir_{k-1}, i = 1,2,3,4,5$; the constrained optimum, $\boldsymbol{x}^* = (0,1,2,-1)^T$.

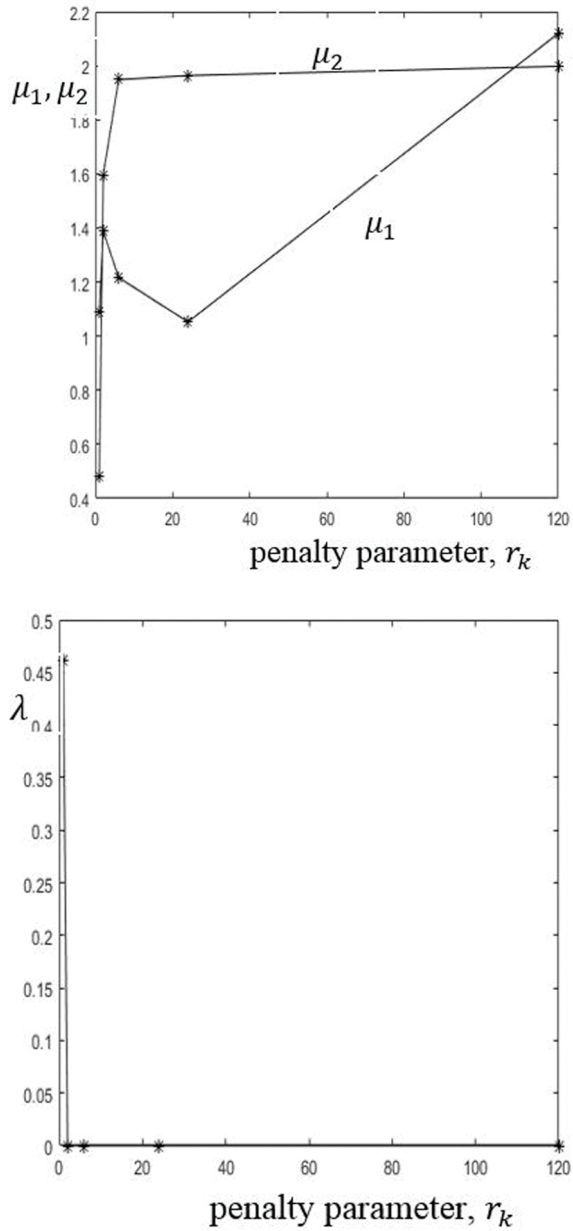


FIGURE 2.17 Augmented Lagrangian method along with CG method; Rosen-Suzuki function (Vanderplaats 1973), evolution of Lagrange multipliers with respect to $r_k = ir_{k-1}, i = 1,2,3,4,5$; (a) multipliers μ_1 and μ_2 corresponding to the equality constraints $h_1(\boldsymbol{x})$ and $h_2(\boldsymbol{x})$ respectively and (b) multiplier λ corresponding to the inequality constraint $g(\boldsymbol{x})$.

■

2.4.4 Sequential Quadratic Programming Method

In the iterative methods presented earlier for constrained optimization, we have used at each iteration one or the other unconstrained optimization method described in Section 2.2. Specifically, Newton's method gives the solution in a single step if the objective function is quadratic. Otherwise, Newton's method iteratively solves the problem using quadratic approximations to $f(\boldsymbol{x})$ at each iteration. This may be regarded as the basis for the sequential quadratic programming (SQP) method [Powell 1978], which we now describe. A constrained problem with a quadratic objective function and linear constraints is called a quadratic programming problem. The SQP method reduces a nonlinear optimization problem into a sequence of such quadratic problems during the iteration process. Suppose that an objective function $f(\boldsymbol{x}) \in C^2(a,b)$ is approximated by a locally quadratic function $\hat{f}(\boldsymbol{x})$ around any $\hat{\boldsymbol{x}}$ using a truncated Taylor series expansion (Section 1.2, Chapter 1):

$$f(\boldsymbol{x}) \cong \hat{f}(\boldsymbol{x}) = f(\hat{\boldsymbol{x}}) + \nabla f(\hat{\boldsymbol{x}})^T (\boldsymbol{x} - \hat{\boldsymbol{x}}) + \frac{1}{2}(\boldsymbol{x} - \hat{\boldsymbol{x}})^T \boldsymbol{H}(\hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}}) \tag{2.130a}$$

$\boldsymbol{H}(\hat{\boldsymbol{x}}) \in \mathbb{R}^{n \times n}$ is the Hessian and $\nabla f(\hat{\boldsymbol{x}}) \in \mathbb{R}^n$ the gradient of $f(\boldsymbol{x})$. Let $h_i, i = 1,2,..,l$ and $g_j, j = 1,2,\ldots,m$ respectively be the given equality and inequality constraints. These constraints, which are generally nonlinear, are linearized as:

$$\mathcal{H}_i(\boldsymbol{x}) = h_i(\hat{\boldsymbol{x}}) + \nabla h_i^T(\hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}}) = 0,\ i = 1,2,\ldots,l$$

$$\mathcal{G}_j(\boldsymbol{x}) = g_j(\hat{\boldsymbol{x}}) + \nabla g_j^T(\hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}}) \le 0, j = 1,2,\ldots,m \tag{2.130b,c}$$

At each iteration, the unconstrained problem is represented by the Lagrangian $L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ (see Equation 1.79 in Chapter 1):

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \hat{f}(\boldsymbol{x}) + \sum_{i=1}^{l} \mu_i \mathcal{H}_i(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j\ \mathcal{G}_j(\boldsymbol{x}) \tag{2.131}$$

$\mu_i, i = 1,2,..,l$ and $\lambda_j, j = 1,2,..,m$ are the Lagrange multipliers. Using the KKT conditions $\nabla_x L = 0$, $\nabla_\mu L = 0$ and $\nabla_\lambda L = 0$ which are all linear, one may obtain the unconstrained optimum, say, by a linear programming (LP) method (to be discussed in the next section of this chapter). Iterations follow with a new quadratic problem formed at each iteration and solved for a refined unconstrained optimum. This process, upon convergence (Schlttkowski 1982), may approach the constrained optimum of the original problem. In this respect, an effective variation may be to use a penalty parameter in forming the Lagrangian (Equation 2.131) similar to the ALM method.

$$L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu},r)=\hat{f}(\boldsymbol{x})+\sum_{i=1}^{l}\mu_i h_i(\boldsymbol{x})+r\sum_{i=1}^{l}{}^{2}(\boldsymbol{x})+\sum_{j=1}^{m}\lambda_j\left(\max\left(g_j(\boldsymbol{x}),-\frac{\lambda_j}{2r}\right)\right)$$

$$+r\sum_{j=1}^{m}\left(\max\left(g_j(\boldsymbol{x}),-\frac{\lambda_j}{2r}\right)\right)^2 \tag{2.132}$$

The introduction of the penalty parameter r enables handling the inequality constraints with ease. Further, the unconstrained optimization problem in Equation (2.132) may be solved by any of the descent methods described in this chapter. If Newton's method is applied, the unconstrained minimum for each quadratic problem (2.132) may be obtained as:

$$\boldsymbol{x}_U^* = \boldsymbol{x}-\left[\nabla_{\boldsymbol{x}}^2 L\right]^{-1}\left\{\nabla_{\boldsymbol{x}} L\right\}\boldsymbol{x} \tag{2.133}$$

$\nabla_{\boldsymbol{x}}^2 L$ is the Hessian and $\nabla_{\boldsymbol{x}} L$ the gradient of the Lagrangian $L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu},r)$. For an expeditious implementation of the method, a finite difference scheme [Hilderbrand 1968] may be employed to numerically obtain $\nabla_{\boldsymbol{x}} L$ in this expression to remain as bold and $\nabla_{\boldsymbol{x}}^2 L$. For instance, for a continuous function $G(\boldsymbol{x})$ with $\nabla G=\left(\frac{\partial G}{\partial x_1},\frac{\partial G}{\partial x_2},\ldots,\frac{\partial G}{\partial x_n}\right)^T$, the first-order derivatives may be obtained by central difference formula as:

$$\frac{\partial G}{\partial x_i}=\frac{G(x_1,\ ..,x_i+\Delta x_i,\ ..,x_n)-G(x_1,\ ..,x_i-\Delta x_i,\ ..,x_n)}{2\Delta x_i} + \mathrm{O}(\Delta x_i^2),\quad i=1,2,\ldots,n \tag{2.134}$$

Similarly, with $\nabla^2 G=\frac{\partial^2 G}{\partial x_i \partial x_j}, 1\le i,j\le n$, the second-order derivatives may be computed as:

$$\frac{\partial^2 G}{\partial x_i^2}=\frac{G(x_1,\ ..,x_i+\Delta x_i,\ ..,x_n)-2G(x_1,\ ..,x_i,\ ..,x_n)+G(x_1,\ ..,x_i-\Delta x_i,\ ..,x_n)}{\Delta x_i^{\,2}} + \mathrm{O}(\Delta x_i^{\,2}) \tag{2.135a}$$

$$\frac{\partial^2 G}{\partial x_i \partial x_j} = \frac{\begin{array}{l} G\left(x_1,\ ..,x_i+\Delta x_i,\ .,x_j+\Delta x_j.,x_n\right)-G\left(x_1,\ ..,x_i+\Delta x_i,\ .,x_j-\Delta x_j.,x_n\right)- \\ G\left(x_1,\ ..,x_i-\Delta x_i,\ .,x_j+\Delta x_j.,x_n\right)+G\left(x_1,\ ..,x_i-\Delta x_i,\ .,x_j-\Delta x_j.,x_n\right) \end{array}}{4\Delta x_i \Delta x_j}$$

$$+ \mathrm{O}\left(\Delta x_i^{\,2}, \Delta x_j^{\,2}\right) \tag{2.135b}$$

O (.) denotes the order of approximation (Section 1.6, Chapter 1). The following is an illustration on the SQP applied to a nonlinear optimization problem.

Example 2.4. We consider Himmelblau function for minimization by SQP.

$$\text{minimize: } f(\boldsymbol{x}) = \left(x_1 + x_2^2 - 7\right)^2 + \left(x_1^2 + x_2 - 11\right)^2$$

$$\text{s.t. } x_1,\ x_2 \geq 0 \tag{2.136}$$

Solution. The Himmelblau function (Figure 2.18) has the following local minima:

case i) $\boldsymbol{x}_1^* = (3,2)^T$ with $f\left(\boldsymbol{x}_1^*\right) = 0$

case ii) $\boldsymbol{x}_2^* = (3.584,-1.848)^T$ with $f\left(\boldsymbol{x}_2^*\right) = 0$

case iii) $\boldsymbol{x}_3^* = (-2.805,3.131)^T$ with $f\left(\boldsymbol{x}_3^*\right) = 0$

case iv) $\boldsymbol{x}_4^* = (-3.779,-3.283)^T$ with $f\left(\boldsymbol{x}_4^*\right) = 0$

The augmented Lagrangian of the quadratic problem at the k^{th} iteration is:

$$L(\boldsymbol{x},\boldsymbol{\lambda},r) = \hat{f}(\boldsymbol{x}) + \lambda_1\left(\max\left(x(1),-\frac{\lambda_1}{2r}\right)\right) + r\left(\max\left(x(1),-\frac{\lambda_1}{2r}\right)\right)^2$$

$$+\lambda_2\left(\max\left(x(2),-\frac{\lambda_2}{2r}\right)\right) + r\left(\max\left(x(2),-\frac{\lambda_2}{2r}\right)\right)^2 \tag{2.137}$$

With $\boldsymbol{x}_0 = (-2,-2)^T$ from an infeasible region, $\lambda_{1,0}, \lambda_{2,0} = (0,0)^T$ and $r = 1$, the solution obtained by SQP is shown in Figures 2.19 and 2.20. Newton's method is utilized to optimize the unconstrained problem in Equation (2.137). $\nabla_x L(\boldsymbol{x})$ and $\nabla_x^2 L(\boldsymbol{x})$ are evaluated at each iteration using the central difference formulae in Equations (2.134) and (2.135).

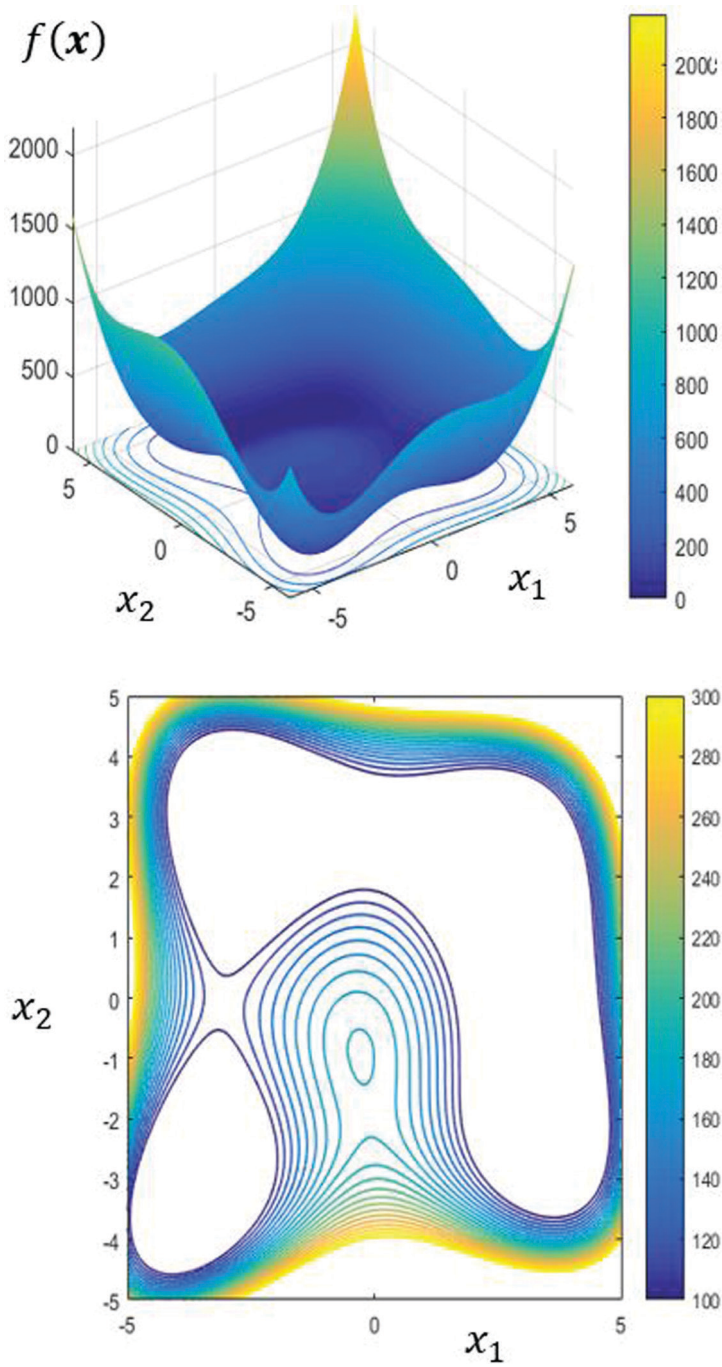


FIGURE 2.18 Himmelblau function; $f(\boldsymbol{x}) = \left(x_1 + x_2^2 - 7\right)^2 + \left(x_1^2 + x_2 - 11\right)^2$: (a) 3D view and (b) planar view.

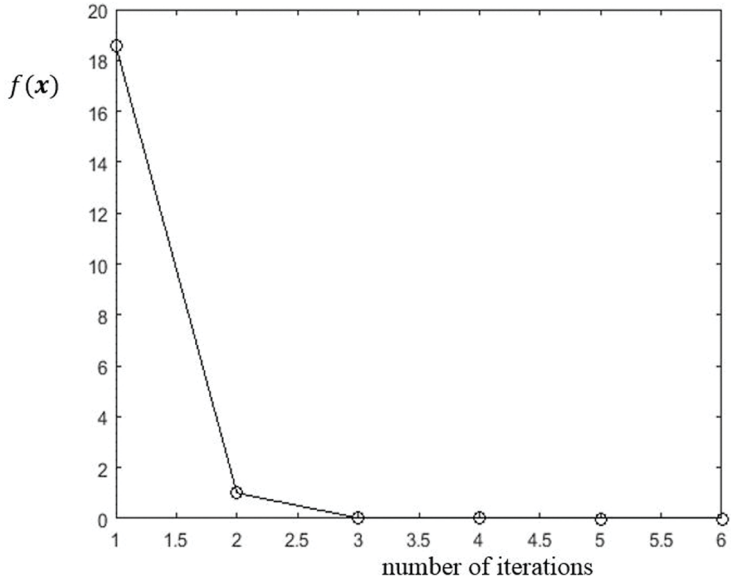


FIGURE 2.19 Optimization by SQP method along with Newton's method, Himmelblau function: $f(\boldsymbol{x}) = \left(x_1 + x_2^2 - 7\right)^2 + \left(x_1^2 + x_2 - 11\right)^2$, evolution of $f(\boldsymbol{x})$ with iterations, $\boldsymbol{x}_0 = (3,\ -1)^T$ and the constrained optimum $\boldsymbol{x}^* = (3,\ 2)^T$ with $f(\boldsymbol{x}^*) = 1.369E-12$.

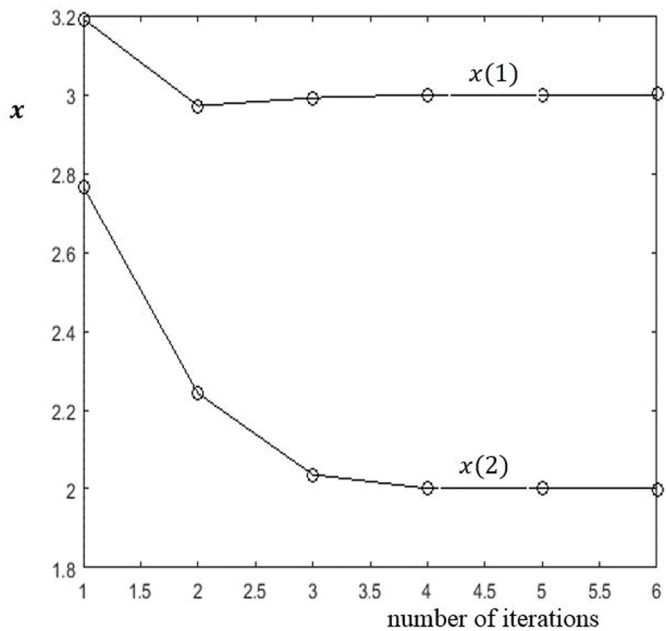
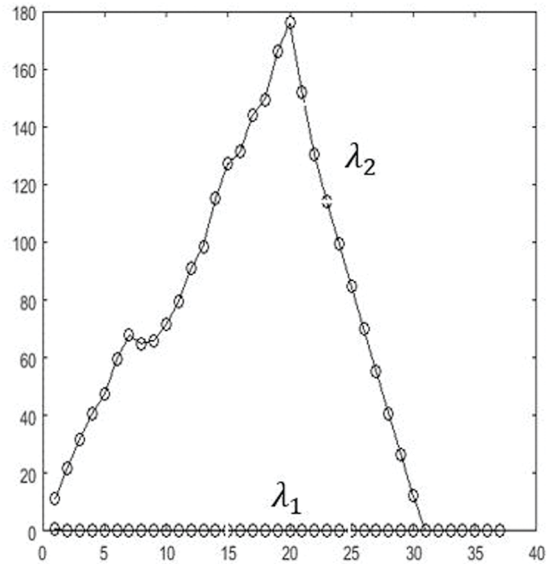


FIGURE 2.20 Optimization of Himmelblau function: $f(\boldsymbol{x}) = \left(x_1 + x_2^2 - 7\right)^2 + \left(x_1^2 + x_2 - 11\right)^2$ by SQP along with Newton's method, evolution of $\boldsymbol{x}^* = (3,\ 2)^T$, case (i) with iterations, $\boldsymbol{x}_0 = (3,\ -1)^T$.



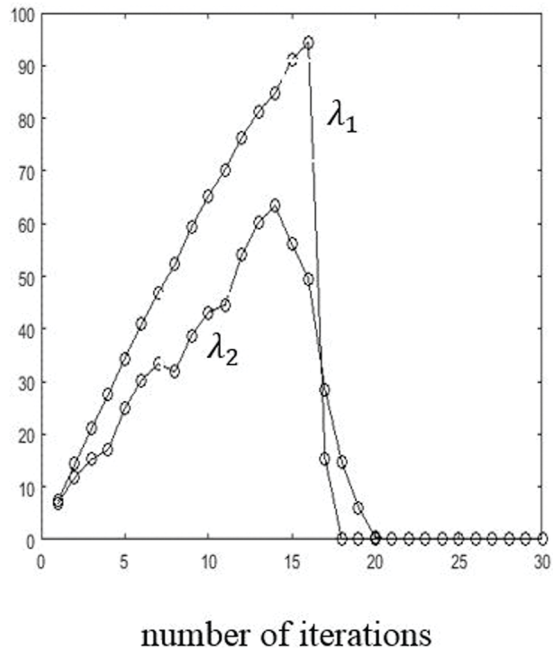


FIGURE 2.21a–b Optimization by SQP method along with Newton's method, Himmelblau function: $f(\boldsymbol{x}) = \left(x_1 + x_2^2 - 7\right)^2 + \left(x_1^2 + x_2 - 11\right)^2$, evolution of λ_1 and λ_2 with iterations, (a) case 2: $\boldsymbol{x}_2^* = \left(3.584,\ -1.848\right)^T$, (b) case 3: $\boldsymbol{x}_3^* = \left(-2.805,\ 3.131\right)^T$ and (c) case 4: $\boldsymbol{x}_4^* = \left(-3.779, -3.283\right)^T$.

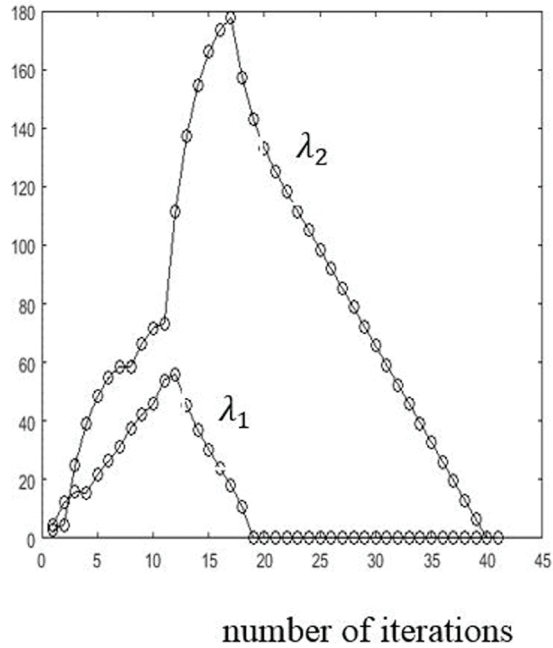


FIGURE 2.21c (Continued)

The other possible local optima can also be obtained with appropriate changes in the inequality constraints. That is, with $g_1(\boldsymbol{x}) = -x_1 \le 0$, and $g_2(\boldsymbol{x}) = x_2 \le 0$, $\boldsymbol{x}_2^* = (3.584, -1.848)^T$ is obtained and with $g_1(\boldsymbol{x}) = x_1 \le 0$, and $g_2(\boldsymbol{x}) = -x_2 \le 0$, $\boldsymbol{x}_3^* = (-2.805, 3.131)^T$ is obtained. The last solution $\boldsymbol{x}_4^* = (-3.779, -3.283)^T$ is obtained with $g_1(\boldsymbol{x}) = x_1 \le 0$, and $g_2(\boldsymbol{x}) = x_2 \le 0$. The starting point $\boldsymbol{x}_0$ is assumed to be the same as $(-2, -2)^T$ in all the four cases. The Lagrange multipliers λ_1 and λ_2 remain inactive during iterations in the first case, i.e. while obtaining $\boldsymbol{x}_1^* = (3,\ 2)^T$. This means that the inequality constraints are always satisfied (with $\lambda_1, \lambda_2 = 0$) at each iteration. For all the other cases, either one of the two or both multipliers are active during the iteration process (see Figure 2.21).

■

2.5 LINEAR PROGRAMMING (LP)

An LP method applies to optimization problems characterized by an objective function and constraints which are linear. In the last section on SQP, a mention is made of solving a quadratic programming problem by LP after linearizing the quadratic objective function and the constraints. While the initial contributions in this field are due to Hitchcock (1941), Kantorovich (1942), and Stigler (1945), it is Dantzig (1951,1963) who is the main architect of the simplex method, a unique way of solving an LP problem. The motivation behind his work was the need for maximal resource

utilization during World War II. Thus, the LP by itself is an innovative and revolutionary development (Hillier and Lieberman 1990) in the history of optimization and is effectively utilized in diverse fields including financial management (Dowling 1991), transportation (Hitchcock 1941, Dantzig and Ramser 1951, Bazaraa *et al.* 1990), operations research and management science (Hillier and Lieberman 1995). Our goal in this section is to present only a brief outline of the LP with main focus on its useful role in solving a general nonlinear optimization problem. A standard LP problem is of the form:

$$\text{minimize } f(\boldsymbol{x}) = \boldsymbol{c}^T \boldsymbol{x}$$

$$\text{s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

$$\boldsymbol{x} \geq 0, \boldsymbol{b} \geq 0 \tag{2.138}$$

$\boldsymbol{c} \in \mathbb{R}^n$ is a column vector, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ a rectangular coefficient matrix and $\boldsymbol{b} \in \mathbb{R}^m$ a column vector, $\boldsymbol{x} \in \mathbb{R}^n$ is the vector of design variables – whose solution is sought for by the LP – and m is the number of constraints of equality and/or inequality type. To bring the inequality constraints into the format of Equation (2.138), we often need slack variables for inequality constraints of 'less than or equal to' (LE) type and surplus variables for those of 'greater than or equal to' (GE) type. Introduction of these variables is meant to transform inequality constraints into the equality (EQ) type. For example, let two inequalities $a_1 x_1 \leq b_1$ and $a_2 x_2 \geq b_2$ be given. The two may be converted to equality constraints of the form:

$$a_1 x_1 + y_1 = b_1 \text{ and}$$

$$a_2 x_2 - y_2 = b_2 \text{ with } y_1, y_2 \geq 0 \tag{2.139}$$

n includes these additional variables too and thus in general, we have $m < n$ in an LP problem. The case with $m = n$ has no necessity for optimization when it possesses a unique solution. The case is of no interest if it has no solution in which case the constraints are inconsistent. When $m > n$, some of the constraints are redundant and may be discarded leading to the case of $m = n$. The following illustration helps in a quick understanding of the fundamentals relevant to the LP formulation and its solution methodology.

Example 2.5. Suppose that two products P_1 and P_2 in a manufacturing unit yield profit as:

$$f(x, y) = 13x + 23y \tag{2.140}$$

Here x and y are the number of manufactured items of P_1 and P_2 respectively. Each product consumes three types of resources, say, a, b and c. The resources are limited by the constraints:

$$5x + 15y \le 480 \text{ –constraint on the resource } a$$

$$4x + 4y \le 160 \text{ –constraint on the resource } b$$

$$35x + 20y \le 1190 - \text{constraint on the resource } c \qquad (2.141)$$

It is required to find the solution for a maximum profit. We must of course have $x \text{ and } y \ge 0$.

Solution. The example is indeed amenable to an easy graphical solution. Figure 2.22 shows the three constraints, the intersecting points (extreme points) of these constraints and the feasible region which is the hatched area *OABCDO* in Figure 2.22a.

The common region bounded by the given constraints is called the feasible region $S = \{\boldsymbol{x} \mid \boldsymbol{Ax} = \boldsymbol{b}, \boldsymbol{x} \ge \boldsymbol{0}\}$. In our two-dimensional case, S is a polygon. In n dimensions, it is known as a polytope. Geometrically, S is the hyperspace formed out of the intersection of the half-spaces defined by the constraints $\boldsymbol{Ax} = \boldsymbol{b}$ and $\boldsymbol{x} \ge \boldsymbol{0}$. Note that the feasible region is convex. That is, if $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are two feasible points

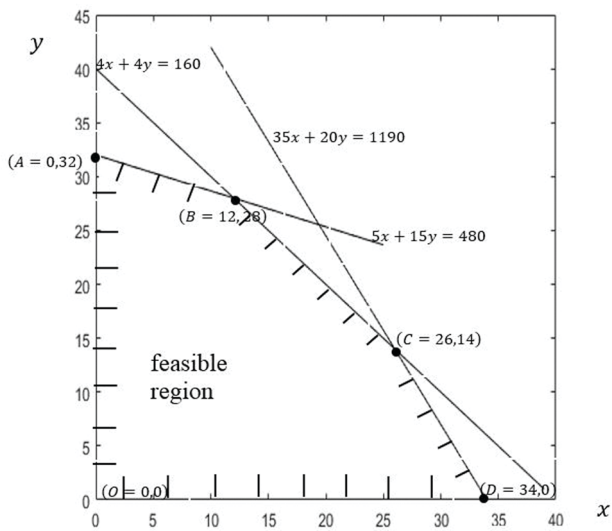


FIGURE 2.22a LP problem in Example 2.5; feasible region shown as the hatched area, constraints shown in dark lines along with extreme points.

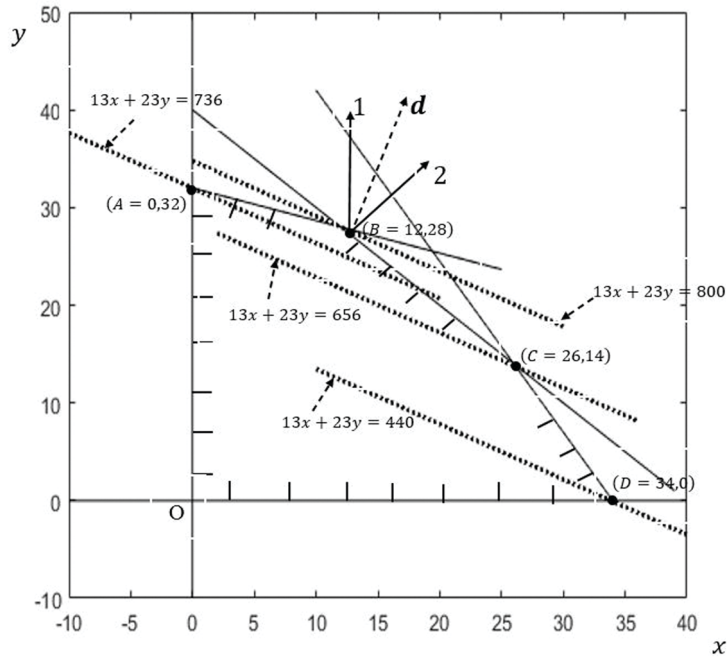


FIGURE 2.22b Graphical solution to LP problem in Example 2.5; dark lines – constraints, dotted lines – equi-potential curves of the objective function passing through the extreme points.

in the feasible region, we find that with $0 \le \lambda \le 1$, $\boldsymbol{x} = \lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2$ is also a feasible point on the line joining $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. A vertex (or a corner point) $\boldsymbol{x}$ in a polytope is an extreme point if $\boldsymbol{x} = \lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2$ with $\boldsymbol{x}_1, \boldsymbol{x}_2 \in S$ and $0 < \lambda < 1$ implies that $\boldsymbol{x} = \boldsymbol{x}_1 = \boldsymbol{x}_2$.

For the example, the value of the objective function f is zero at the origin. The objective function values at the other vertices A, B, C and D are 736, 800, 660 and 442 respectively. The equi-potential curves passing through these vertices are shown in Figure 2.22b in dotted lines. As we move from the origin towards the vertices A, B, C and D, the objective function attains a maximum $\boldsymbol{x}^*$ at the vertex B, which is obviously the optimal solution. At the vertex B, the two constraints $4x + 4y \le 160$ and $5x + 15y \le 480$ are found to be active and binding. The gradient vectors (normals) to these active constraints are marked '1' and '2' in Figure 2.22b. The gradient $\nabla f(\boldsymbol{x}^*)$ at B, which points towards the steepest ascent direction $\boldsymbol{d}$, lies within the sector formed by the normals '1' and '2'. In the general n-dimensional case, this implies that the convex cone spanned by the outer normals to the binding constraints contains $\boldsymbol{d}$. This is the requirement of KKT conditions for optimality (Section 1.6.3, Chapter 1). Solutions by such graphical means, though insightful in the 2D case, may not be feasible for more than two dimensions. The simplex method

(Dantzig 1963, Wolfe 1959) is a powerful computational technique for solving LP problems as in (2.138) with a large number of design variables. In Example 2.5, the inequality constraints are of LE type. They may be written in the standard form by adding slack variables as:

$$5x+15y+z=480-\text{constraint on resource } a$$

$$4x+4y+w=160-\text{constriant on resource } b$$

$$35x+20y+v=119-\text{constriant on resource } c \tag{2.142a}$$

z, w and v are non-negative slack variables. With the introduction of these slack variables, $\boldsymbol{x}=(x,y,z,w,v)^T$. Here $n=5$ and $m=3$. We also have:

$$\boldsymbol{A}=\begin{bmatrix}5 & 15 & 1 & 0 & 0\\ 4 & 4 & 0 & 1 & 0\\ 35 & 20 & 0 & 0 & 1\end{bmatrix}=\begin{bmatrix}A_n I_b\end{bmatrix} \text{ with}$$

$$A_n=\begin{bmatrix}5 & 15\\ 4 & 4\\ 35 & 20\end{bmatrix} \text{ and } I_b=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$\boldsymbol{b}=\begin{pmatrix}480 & 160 & 1190\end{pmatrix}^T \tag{2.142b}$$

The m constraint equations $\boldsymbol{Ax}=\boldsymbol{b}$ may be written in the following partitioned form:

$$\boldsymbol{A}_n\boldsymbol{x}_n+\boldsymbol{I}_b\boldsymbol{x}_b=\boldsymbol{b} \tag{2.143}$$

$\boldsymbol{x}_n=(x,y)^T$ and $\boldsymbol{x}_b=(z,w,v)^T$. $\boldsymbol{I}_b$ is a $m\times m$ non-singular sub-matrix of $\boldsymbol{A}$ consisting of linearly independent columns. Note that the subscript 'b' is in no way related the vector $\boldsymbol{b}$ in Equation (2.143). The original constraint equations recast in the standard form readily yields one possible solution $\boldsymbol{x}=(0,0,480,160,1190)^T$. One such solution is known as the basic solution which is obtained by setting variables x and y to zero and solving $\boldsymbol{Ax}=\boldsymbol{b}$ for the remaining variables z,w and v. That is:

$$\boldsymbol{x}_b=\boldsymbol{I}_b^{-1}\left(\boldsymbol{b}-\boldsymbol{A}_n\boldsymbol{x}_n\right) \tag{2.144}$$

This is possible since $\boldsymbol{A}$ is in a canonical form[‡‡] and the rank of $\boldsymbol{A}$ is equal to m. The variables $\boldsymbol{x}_b=(z,w,v)=(480,160,1190)\in\mathbb{R}^m$ are called the basic variables

[‡‡] *Canonical form of a matrix*
It is a unique representation of a matrix. The triangular form, Jordan canonical form (not covered in this book) and row echelon form are some of the canonical forms for a matrix.

with respect to the bases which are the columns of I_b. The remaining variables $\boldsymbol{x}_n = (x, y) = (0,0) \in \mathbb{R}^{n-m}$ are known as the non-basic variables. The basic solution is a basic feasible solution if all the basic variables satisfy the non-negativity requirement. $\boldsymbol{x}_b$ here is a basic feasible solution. A basic feasible solution which optimizes $f(\boldsymbol{x})$ is the optimal basic feasible solution. Note that each vertex or the extreme point in an n-dimensional polytope is a basic feasible solution. This is since a polytope is a closed convex set (Dantzig 1963). Here the word 'closed' is a qualifier to the convex set due to the equality sign in LE or GE type of inequalities holding good. The optimum occurs only at one of the extreme points. Proof for this fundamental result in an LP is as follows.

Proof: Since the feasible region S is a convex polytope, it is non-empty, closed and bounded. Let $\boldsymbol{x}_i, i = 1,2,\ldots,K$ be the K extreme points on S. Any other point $\boldsymbol{x} \in S$ may be expressed as a convex combination of the extreme points:

$$\boldsymbol{x} = \sum_{i=1}^{K} a_i \boldsymbol{x}_i, a_i \geq 0 \text{ and } \sum_{i=1}^{K} a_i = 1 \tag{2.145}$$

If $\boldsymbol{x}^* \in \{\boldsymbol{x}_i, i = 1,2,\ldots,K\}$ such that $f(\boldsymbol{x}^*) = \max\{f(\boldsymbol{x}_i), i = 1,2,\ldots,K\}$, then one has:

$$f(\boldsymbol{x}) = \boldsymbol{c}^T \boldsymbol{x} = \boldsymbol{c}^T \sum_{i=1}^{K} a_i \boldsymbol{x}_i$$

$$= \sum_{i=1}^{K} a_i (\boldsymbol{c}^T \boldsymbol{x}_i)$$

$$= \sum_{i=1}^{K} a_i f(\boldsymbol{x}_i)$$

$$\leq \sum_{i=1}^{K} a_i f(\boldsymbol{x}^*)$$

$$\leq f(\boldsymbol{x}^*) \sum_{i=1}^{K} a_i$$

$$\leq f(\boldsymbol{x}^*) \tag{2.146}$$

Therefore, $f(\boldsymbol{x}) \leq f(\boldsymbol{x}^*)$ for any $\boldsymbol{x} \in S$ and the proof is complete.

♦

For a large LP problem, searching for an optimum among all the possible basic solutions may not be an easy task. In fact, for a matrix of size $n \times m$, the number of possible basic solutions is n_{cm} which may be an exceedingly large number even for relatively small n and m. The simplex method is an iterative technique that proceeds with a known basic feasible solution (one extreme point) to the next basic feasible solution (another extreme point). Therefore, the method examines the set of extreme points in the convex set, finally yielding the optimal feasible solution in a finite number of steps. Referring back to the example for which the initial basic feasible solution $\boldsymbol{x} = (0,0,480,160,1190)^T$, the following simplex Tableau shows the result.

The last row in the above tableau corresponds to the objective function $-f = -\boldsymbol{c}^T\boldsymbol{x} = -13x - 23y$. Here the maximization of f is written as an equivalent of the minimization of $-f$. With $x, y = 0$ in the basic solution, the initial value of $-f$ is zero (shown in the last row and the last column). This corresponds to the extreme point O, the origin in Figure 2.22b. The next task in the simplex method is to examine the elements of vector $\boldsymbol{c}$ in the last row of the tableau and to pick a non-basic variable $x_{n,j}$ for which the coefficient c_j is negatively maximum. The coefficient $c_2 = -23$ (enclosed within a box in the last row of the tableau) corresponding to the non-basic variable y is most negative. The value of $-f$ gets reduced if y is increased from the present zero value to any positive quantity while keeping the other non-basic variable x at zero. This criterion chooses y to enter the basic variable set from the nonbasic variable set. We simultaneously move a basic variable among $\boldsymbol{x}_b = (z, w, v)$ to the nonbasic variable set $\boldsymbol{x}_n$ in place of y. This is decided by an observation of the elements in the column vector $\boldsymbol{A}_{n,j} = (15 \;\; 4 \;\; 20)^T$ with $j = 2$ corresponding to y. While y may be made as large as possible, a feasibility requires the basic variables $\boldsymbol{x}_b$ to remain non-negative. From Table 2.4, the corresponding values of the basic variables in terms of y are:

$$z = 480 - 15y$$

$$w = 160 - 4y$$

$$v = 1190 - 20y \tag{2.147}$$

The largest possible value to which y may be increased is the minimum of the ratios 480/15, 160/4 and 1190/20. The minimum ratio is $\dfrac{480}{15} = 32$, corresponding to the row of the non-basic variable z. Hence z leaves $\boldsymbol{x}_b$ giving way to y. The new $\boldsymbol{x}_b$ is $(y \;\; w \;\; v)$ and the new $\boldsymbol{x}_n$ is $(x \;\; z)$. By pivoting operation on the columns of $\boldsymbol{A}$, the columns corresponding to the new $\boldsymbol{x}_b$ are brought to the form of $\boldsymbol{I}_b$. This gives the new basic solution as $y = 32$, $w = 32$, $v = 550$ as shown in Table 2.5. The objective function value $-f$ reduces from earlier zero to -736 corresponding to the extreme point A in Figure 2.22b.

Following similar arguments for the next iteration, one seeks for a possible new non-basic variable to enter $\boldsymbol{x}_b$ in place of an existing one. This attempt finds x

TABLE 2.4
LP Problem in Example 2.5 and Simplex Method Tableau at Zeroth Iteration

	Original variables		Slack variables			b_i (RHS
Basic variables	***x***	***y***	***z***	***y***	***v***	**elements)**
z	5	15	1	0	0	480
w	4	4	0	1	0	160
v	35	20	0	0	1	1190
–f	–13	–23	0	0	0	0

TABLE 2.5
LP Problem in Example 2.5 and Simplex Method Tableau at the First Iteration

Basic	Original variables		Slack variables			b_i (RHS
variables x_b	***x***	***y***	***z***	***w***	***v***	**elements)**
z	$\frac{1}{3}$	1	$\frac{1}{15}$	0	0	32
w	$\frac{8}{3}$	0	$-\frac{4}{15}$	1	0	32
v	$\frac{85}{3}$	0	$-\frac{4}{3}$	0	1	550
–f	–16/3	0	23/15	0	0	–736

replacing w in $\boldsymbol{x}_b$ and w entering $\boldsymbol{x}_n$. Table 2.6 shows the result at the end of the 2nd iteration which is indeed the final step giving $-f = -800$ with $x = 12$ and $y = 28$. This is the optimal basic feasible solution corresponding to the extreme point B in Figure 2.22b.

The stopping criterion for the simplex method is that no more entries in the last row corresponding to the objective function are negative. It implies that no more positive increase in any of the design variables has an effect on further minimization of $f(\boldsymbol{x})$.

The following example shows the simplex method applied to a nonlinear optimization problem via SQP. It also illustrates the way to handle equality constraints in an LP. This type of constraint may require the introduction of additional artificial variables so as to bring the matrix $\boldsymbol{A}$ into a canonical form. Note that the canonical form helps in quickly picking up a basic feasible solution required to initiate the simplex method. Otherwise, one needs to execute pivoting operations to reduce $\boldsymbol{A}$ to the canonical form.

TABLE 2.6
LP Problem in Example 2.5 and Simplex Method Tableau at Second Iteration

Basic variables	Original variables		Slack variables			b_i (RHS elements)
	x	y	z	w	v	
y	0	1	$\frac{1}{10}$	$-\frac{1}{8}$	0	28
x	1	0	$-\frac{1}{10}$	$\frac{3}{8}$	0	12
v	0	0	$\frac{3}{2}$	0	1	210
$-f$	0	0	1	2	0	–800

Example 2.6. We refer to the problem in Example 2.4 of minimizing the Himmelblau function $f(\boldsymbol{x}) = \left(x_1 + x_2^2 - 7\right)^2 + \left(x_1^2 + x_2 - 11\right)^2$. The task is to find the optimum solution by applying the simplex method to the linear KKT conditions derived from an equivalent SQP to $f(\boldsymbol{x})$.

Solution. Equation (2.130a) gives a quadratic approximation $\hat{f}(\boldsymbol{x})$ to $f(\boldsymbol{x})$ around any $\hat{\boldsymbol{x}}$. The gradient vector ∇f and the Hessian matrix $\boldsymbol{H}$ are:

$$\nabla f = \begin{pmatrix} 2\left(x_1 + x_2^2 - 7\right) + 4x_1\left(x_1^2 + x_2 - 11\right) \\ 4x_2\left(x_1 + x_2^2 - 7\right) + 2\left(x_1^2 + x_2 - 11\right) \end{pmatrix}$$

$$\boldsymbol{H} = \begin{bmatrix} 12x_1^2 + 4x_2 - 42 & 4\left(x_1 + x_2\right) \\ 4\left(x_1 + x_2\right) & 12x_2^2 + 4x_1 - 26 \end{bmatrix} \tag{2.148}$$

In terms of $\boldsymbol{H}$ and $\nabla f(\hat{\boldsymbol{x}})$, the objective function is:

$$\hat{f}(\boldsymbol{x}) = \nabla f^T(\boldsymbol{x} - \hat{\boldsymbol{x}}) + \frac{1}{2}(\boldsymbol{x} - \hat{\boldsymbol{x}})^T \boldsymbol{H}(\boldsymbol{x} - \hat{\boldsymbol{x}}) \tag{2.149}$$

$f(\hat{\boldsymbol{x}})$ which is a part of $\hat{f}(\boldsymbol{x})$ (Equation 2.130a) is ignored in finding the optimum since it is a constant. One needs to minimize $\hat{f}(\boldsymbol{x})$ under the constraint $\boldsymbol{x} \geq 0 \equiv -\boldsymbol{x} \leq 0$. Employing the Lagrange multipliers $\boldsymbol{\lambda} = \left(\lambda_1, \lambda_2\right)^T$ to the inequality constraint, one gets the following KKT conditions for optimality (Equation 1.53 in Chapter 1):

$$\nabla L_{\boldsymbol{x}} = \nabla \hat{f}_{\boldsymbol{x}} - \boldsymbol{x}$$

$$\Rightarrow \nabla f + \boldsymbol{H}^T \left(\boldsymbol{x} - \hat{\boldsymbol{x}}\right) - \boldsymbol{\lambda} = 0$$

$$\Rightarrow \boldsymbol{H}^T \boldsymbol{x} - \boldsymbol{\lambda} = -\nabla f + \boldsymbol{H}^T \hat{\boldsymbol{x}} = \boldsymbol{b} \tag{2.150a}$$

$$\boldsymbol{x} \geq 0 \tag{2.150b}$$

$$\boldsymbol{\lambda} \geq 0 \tag{2.150c}$$

And finally, the complementary slackness condition:

$$\boldsymbol{\lambda x} = 0 \tag{2.150d}$$

Equations (2.150b–c) indicate the non-negativity requirement for the variables $\boldsymbol{x}$ and $\boldsymbol{\lambda}$. Note that while the KKT conditions (2.150a–c) are all linear, the complementary slackness condition (2.150d) is nonlinear. However, the condition can be accommodated in the simplex method as a qualitative constraint, i.e., by ensuring that an active constraint i.e., $x_i = 0$ has non-zero multiplier λ_i and vice-versa. That is, when one of the pair (λ_i, x_i) is in the basis set, it is required that the other one is in the non-basis set. With this LP setting, a quadratic approximation around $\hat{\boldsymbol{x}} = (2,1)^T$ has the KKT conditions:

$$\begin{bmatrix} 10 & 12 & -1 & 0 \\ 12 & -6 & 0 & -1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 88 \\ 46 \end{pmatrix} \tag{2.151}$$

The last equation is in the form of equality constraints. The simplex method needs a basic feasible solution to start the iterative process. A basic feasible solution may be readily obtained by introducing artificial variables $\boldsymbol{y} = (y_1, y_2)^T$ into the two equations with $y_1, y_2 \geq 0$. This renders the coefficient matrix transformed into a canonical form as:

$$\begin{bmatrix} 10 & 12 & -1 & 0 & 1 & 0 \\ 12 & -6 & 0 & -1 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \lambda_1 \\ \lambda_2 \\ y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 88 \\ 46 \end{pmatrix} \tag{2.152}$$

TABLE 2.7
LP Problem in Example 2.6 and Simplex Method Tableau at 2nd (final) Iteration of the First Quadratic Approximation at $x = (2,1)^T$

Basic variables	Original variables		Lagrange multipliers		Additional variables		b_i (RHS elements)
	x_1	x_2	λ_1	λ_2	y_1	y_2	
x_1	0	1	$-\frac{1}{17}$	$\frac{5}{102}$	$\frac{1}{17}$	$-\frac{5}{102}$	2.9216
x_2	1	0	$-\frac{1}{34}$	$-\frac{1}{17}$	$\frac{1}{34}$	$\frac{1}{17}$	5.2941
f	0	0	0	0	1	1	0

To use the simplex method, we introduce a linear objective function f as a sum of the two artificial variables y_1 and y_2. That is, we minimize:

$$f = y_1 + y_2$$

$$\Rightarrow f = 134 - 22x_1 - 6x_2 + \lambda_1 + \lambda_2 \tag{2.153}$$

The last expression for f is obtained from Equation (2.152) by expressing y_1 and y_2 in terms of the other variables. Now, with the initial feasible solution $y_1 = 88$ and $y_2 = 46$ and $f = 134$, the simplex method is applied to obtain the optimal solution. The stopping criterion for the method is to render f to a zero value with y_1 and y_2 relegated to the non-basic set where the two variables are simultaneously zero. Table 2.7 shows the optimal basic feasible solution $\boldsymbol{x} = (2.9216,\ 5.2941)^T$ for the first quadratic approximation to $f(\boldsymbol{x})$ at $\hat{\boldsymbol{x}} = (2,1)^T$.

With the solution vector $\boldsymbol{x}$ obtained at the end of each approximation, the quadratic approximation to $f(\boldsymbol{x})$ is reformulated and the simplex method repeated. Figure 2.23 shows the final result reaching the optimum $\boldsymbol{x}^* = (3,2)^T$ at the end of ten quadratic approximations to $f(\boldsymbol{x})$.

■

Note that one may use artificial variables in an LP problem where a regular objective function is already present. A two phase method is generally employed in such a case. With the readily available starting solution in terms of the artificial variables, simplex method is initiated in phase I to obtain a feasible solution in terms of the original design variables. This is carried out by minimizing the sum of the artificial variables as in the last Example 2.6. Using this feasible solution (with the artificial variables discarded) as a starting point, phase II applies the simplex method to minimize the original objective function.

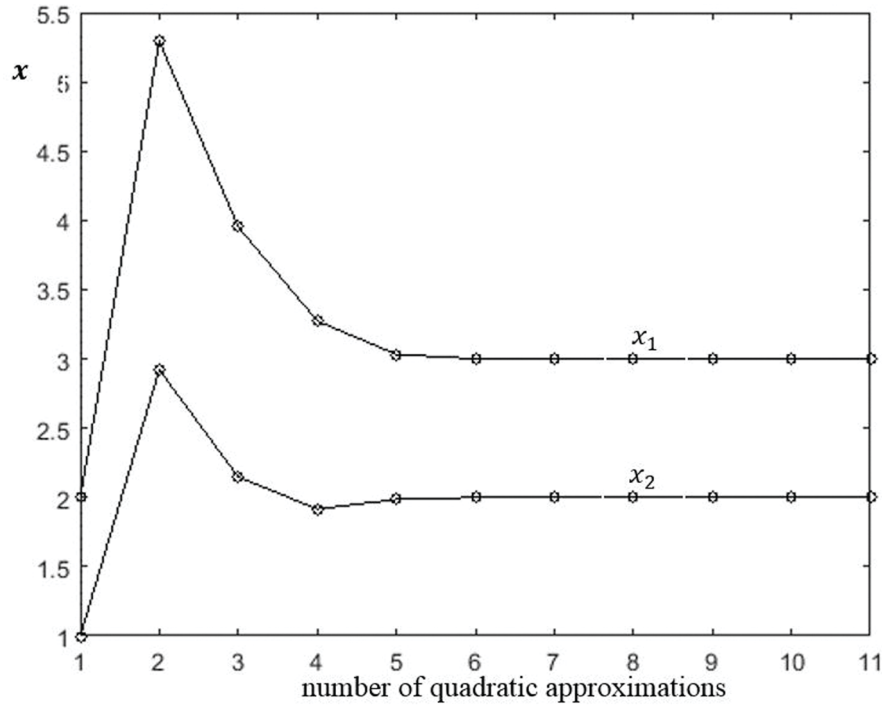


FIGURE 2.23 Solution to Example 2.6 by simplex method via SQP, optimum $\boldsymbol{x}^* = (3,2)^T$ (see the result in Figure 2.20 obtained by SQP plus Newton's method).

A bountiful of literature is available (Murty 1983, Hillier and Lieberman 1995, Gärtner and Matoušek 2006) on LP and its wide usage in industry for solving large scale optimization problems. Also, many algorithms have progressively emerged to improve the computational efficiency of the simplex method. The revised simplex (Dantzig and Thapa 2003), interior point (Karmarkar 1984, Roos *et al.* 2006) and dual simplex methods (Lemke 1954, Florian *et al.* 1981, Goldfarb 1985) are such extensions to the original simplex. In particular, the interior point method of Karmarkar (1984) is a landmark development in LP and is polynomial (Appendix B) in computing time. Dual simplex method solves an LP problem by transforming it into its dual. Solving a dual problem may be advantageous in certain cases particularly when the number of design variables is far less than the number of constraints.

2.6 METHOD OF GENERALIZED REDUCED GRADIENTS

In the context of nonlinear optimization, the method of generalized reduced gradients (GRG) mimics the LP problem. As with the LP problem, the solution vector $\boldsymbol{x} \in \mathbb{R}^n$ is divided into two components – vectors of basic and non-basic variables. If slack variables are introduced to transform the inequality constraints to equality constraints, $\boldsymbol{x}$ contains both the original and these slack variables. As developed by Wolfe (1963),

the method applies to a nonlinear optimization problem with linear constraints. It has been extended to problems with nonlinear constraints by Abadie and Carpenter (1969). Let us define a nonlinear optimization problem as:

$$\text{Minimize } f(\boldsymbol{x})$$

$$\text{s.t. } h_i(\boldsymbol{x}) = 0, i = 1,2,\ldots,m \text{ and}$$

$$\boldsymbol{x}_l \le \boldsymbol{x} \le \boldsymbol{x}_u \tag{2.154}$$

$\boldsymbol{x}_l$ and $\boldsymbol{x}_u$ are the lower and upper bounds for $\boldsymbol{x}$. Assume that the nonlinear objective and constraint functions are differentiable. Using the equality constraints, we express m of the variables in $\boldsymbol{x}$ in terms of the remaining $n-m$ variables. $\boldsymbol{x}$ is thus split as $\left(\boldsymbol{x}_b^T, \boldsymbol{x}_{nb}^T\right)$. $\boldsymbol{x}_b \in \mathbb{R}^m$ is called the vector of basic (dependent) variables and $\boldsymbol{x}_{nb} \in \mathbb{R}^{n-m}$ the vector of non-basic (independent) variables as in the LP case. Thus, if $\hat{\boldsymbol{x}}$ is a feasible point during the iteration process, expressing $\boldsymbol{x}_b$ in terms of $\boldsymbol{x}_{nb}$ is possible, say, in the neighbourhood of $\hat{\boldsymbol{x}}$ by the implicit function theorem. The theorem requires that $\left[\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_b}\right] \in \mathbb{R}^{m\times m}$ is non-singular for all $\boldsymbol{x}_b$ in the above neighbourhood. $\left[\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_b}\right]$ at $\hat{\boldsymbol{x}}$ is an $m \times m$ matrix and may be reduced to an identity matrix by pivoting as in the simplex method (see the definition of matrix $\boldsymbol{A}_b$ in Equation 2.142b). It follows that the optimization problem in Equation (2.154) reduces to:

$$\text{minimize } \hat{f}(\boldsymbol{x}_{nb})$$

$$\text{s. t. } \boldsymbol{x}_{l,nb} \le \boldsymbol{x}_{nb} \le \boldsymbol{x}_{u,nb} \tag{2.155}$$

where $\hat{f}(\boldsymbol{x}_{nb}) = f\left(\boldsymbol{x}_b(\boldsymbol{x}_{nb}), \boldsymbol{x}_{nb}\right)$ and $\boldsymbol{x}_{l,nb} \subset \boldsymbol{x}_l$ and $\boldsymbol{x}_{u,nb} \subset \boldsymbol{x}_u$ are respectively the lower and upper limits for the non-basic variables. GRG aims at iteratively solving a sequence of such reduced dimensional unconstrained problems as defined in the last equation. Writing $\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} = \left[\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_b} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_{nb}}\right]$, the differential $d\boldsymbol{h}$ is given by:

$$d\boldsymbol{h} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_b} d\boldsymbol{x}_b + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_{nb}} d\boldsymbol{x}_{nb} = \boldsymbol{B} d\boldsymbol{x}_b + \boldsymbol{C} d\boldsymbol{x}_{nb} \tag{2.156a}$$

Here $\boldsymbol{B} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{C} \in \mathbb{R}^{m\times(n-m)}$. With the equality constraints active, i.e. with $\boldsymbol{h}\left(\boldsymbol{x}_b, \boldsymbol{x}_{nb}\right) = \boldsymbol{0}$ in the neighbourhood of $\hat{\boldsymbol{x}}$, one has:

$$d\boldsymbol{h} = \boldsymbol{0} \Rightarrow d\boldsymbol{x}_b = -\boldsymbol{B}^{-1}\boldsymbol{C}d\boldsymbol{x}_{nb} \tag{2.156b}$$

Similarly, if we write $df = \left(\frac{\partial f}{\partial \boldsymbol{x}_b}\right)^T d\boldsymbol{x}_b + \left(\frac{\partial f}{\partial \boldsymbol{x}_{nb}}\right)^T d\boldsymbol{x}_{nb}$, then

$$df = -\left(\frac{\partial f}{\partial \boldsymbol{x}_b}\right)^T \boldsymbol{B}^{-1}\boldsymbol{C}d\boldsymbol{x}_{nb} + \left(\frac{\partial f}{\partial \boldsymbol{x}_n}\right)^T d\boldsymbol{x}_{nb} \qquad \text{(from Equation 2.156b)}$$

$$= \left[-\left(\frac{\partial f}{\partial \boldsymbol{x}_b}\right)^T \boldsymbol{B}^{-1}\boldsymbol{C} + \left(\frac{\partial f}{\partial \boldsymbol{x}_{nb}}\right)^T\right] d\boldsymbol{x}_{nb}$$

$$\Rightarrow \frac{df}{d\boldsymbol{x}_{nb}} = \left[-\left(\frac{\partial f}{\partial \boldsymbol{x}_b}\right)^T \boldsymbol{B}^{-1}\boldsymbol{C} + \left(\frac{\partial f}{\partial \boldsymbol{x}_{nb}}\right)^T\right]^T \tag{2.157}$$

$\frac{df}{d\boldsymbol{x}_{nb}}$ is called the reduced gradient in $n-m$ dimensional space. It may be denoted by $\frac{d\hat{f}}{d\boldsymbol{x}_{nb}}$ and used to optimize $\hat{f}\left(\boldsymbol{x}_{nb}\right)$ in the neighbourhood of $\hat{\boldsymbol{x}}_{nb} \subset \hat{\boldsymbol{x}}$. With an optimum $\boldsymbol{x}_{nb}$ so obtained, we proceed to the next iteration to formulate and solve another such reduced dimensional optimization problem. The procedure is repeated to finally realize the optimum $\boldsymbol{x}^*$ for the original problem. However, obtaining $\hat{f}\left(\boldsymbol{x}_{nb}\right)$ from $f\left(\boldsymbol{x}\right)$ in an explicit form at each iteration may be difficult. Using the reduced gradient in Equation (2.157), computations may be conveniently carried out in the original format of n variables itself (Rao 1984, Belegundu and Chandrupatla 1999, Bazaraa *et al.* 2006). The following illustration explains the iterative steps in GRG with equality constraints.

Example 2.7. Consider optimization of the following function (Rao 1996) with $\boldsymbol{x} = \left(x_1, x_2, x_3\right)^T$:

$$\text{minimize } f(\boldsymbol{x}) = (x_1 - x_2)^2 + (x_2 - x_3)^4$$

s. t.

$$h(\boldsymbol{x}) = x_1(1 + x_2^2) + x_3^4 - 3 = 0$$

and $x_l(i) = -3 \le x_1, x_2, x_3 \le 3 = x_u(i)$,

$$i = 1, 2, 3. \tag{2.158}$$

Solution. Here $n = 3$ and $m = 1$. Let us start the iterations with the feasible point $\hat{\boldsymbol{x}} = (-2.6, 2, 2)^T$ and $h(\hat{\boldsymbol{x}}) = 0$. The starting value of $f(\hat{\boldsymbol{x}}) = 21.16$. The gradient vector ∇_x is:

$$\nabla_x f = \frac{\partial f}{\partial \boldsymbol{x}} = \left(2(x_1 - x_2), -2(x_1 - x_2) + 4(x_2 - x_3)^3, -4(x_2 - x_3)^3\right)^T \tag{2.159}$$

and

$$\nabla_x h = \left[(1 + x_2^2),\ 2x_1x_2,\ 4x_3^3\right] \tag{2.160}$$

Iterations start with the selection of the basic and non-basic variables. Here, x_3 is chosen to be the basic variable for all iterations. Otherwise, selection of the basic variables is usually made via pivoting operation on $\boldsymbol{A} = \nabla_x h$.

TABLE 2.8
Gradient Vectors of the Objective Function and Constraints at the Feasible Point $\hat{x} = (-2.6,\ 2,\ 2)^T$ at the Start of the First Iteration

	$\boldsymbol{B} = \dfrac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_b} = \nabla_{x_b} \boldsymbol{h}$	$\boldsymbol{C} = \dfrac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_{nb}} = \nabla_{x_{nb}} \boldsymbol{h}$
	Basic variables	**Non-basic variables**
$\nabla_x \boldsymbol{h}$	$\boldsymbol{x}_b = \boldsymbol{x}_3$	$\boldsymbol{x}_{nb} = (\boldsymbol{x}_1, \boldsymbol{x}_2)^T$
$\boldsymbol{A} = [B \quad C]$	$B = [32]$	$\boldsymbol{C} = [5 \quad -10.4]^T$
$\nabla_x f$	$\dfrac{\partial f}{\partial \boldsymbol{x}_b} = \nabla_{x_b} f = [0]$	$\dfrac{\partial f}{\partial \boldsymbol{x}_{nb}} = \nabla_{x_n} f$ $= [-9.2,\ 9.2]^T$

Table 2.8 shows these variables along with the vector $\nabla_x f$ and the matrix $\boldsymbol{A}$ evaluated at $\boldsymbol{x} = \hat{\boldsymbol{x}}$.

The reduced gradient from Equation (2.157) is:

$$\frac{d\hat{f}}{d\boldsymbol{x}_{nb}} = \left[-\left(\frac{\partial f}{\partial x_b} \right)^T B^{-1}\boldsymbol{C} + \left(\frac{\partial f}{\partial \boldsymbol{x}_{nb}} \right)^T \right]^T = \left(-9.2,\ 9.2\right)^T \tag{2.161}$$

The steepest descent direction $d\boldsymbol{x}_{nb}$ for $\boldsymbol{x}_{nb}$ is $-\dfrac{d\hat{f}}{d\boldsymbol{x}_{nb}}$. Equation (2.156b) gives the descent direction dx_b for the basic variables as $dx_b = -B^{-1}\boldsymbol{C}d\boldsymbol{x}_{nb} = -4.4275$. To proceed with the unconstrained optimization in the original scenario of n variables, we use $d = \left(d_{x_b}^T \quad d_{x_{nb}}^T \right)^T$ for a line search. With the specified lower and upper bounds $\boldsymbol{x}_l$ and $\boldsymbol{x}_u$ for $\boldsymbol{x}$ in Equation (2.158), it is convenient to fix a suitable step size $s > 0$. That is, with $\boldsymbol{x}_0 = \hat{\boldsymbol{x}}$, the maximum step size s_{nb} for the non-basic variables with respect to $\boldsymbol{x}_{l,nb}$ and $\boldsymbol{x}_{u,nb}$ may be fixed as:

$$s_{nb} = \frac{x_{l,nb}(i) - x_{0,nb}(i)}{d_{nb}(i)}, \text{ if } dx_{nb}(i) < 0 \text{ and}$$

$$s_{nb} = \frac{x_{u,nb}(i) - x_{0,nb}(i)}{d_{nb}(i)}, \text{ if } dx_{nb}(i) > 0 \tag{2.162}$$

Similarly the maximum step size s_b for the basic variables is estimated. The smaller of s_b and s_{nb} may be taken as the maximum step size s_{max} and the update is obtained with $d_0 = d$ as:

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + s_{max}\boldsymbol{d}_0 \tag{2.163}$$

The update is accepted for the next iteration provided $f(\boldsymbol{x}_1) < f(\boldsymbol{x}_0)$. In the present problem, $\min(s_b, s_{nb}) = 0.5435$. Thus, with $s_{min} = 0.5435$, $\boldsymbol{x}_1$ as obtained from (2.163) gives $f(\boldsymbol{x}_1) = 74.42 > f(\boldsymbol{x}_0)$. The update is obviously not acceptable and a line search is performed to find a suitable step size $s \in [0, s_{max}]$ such that an

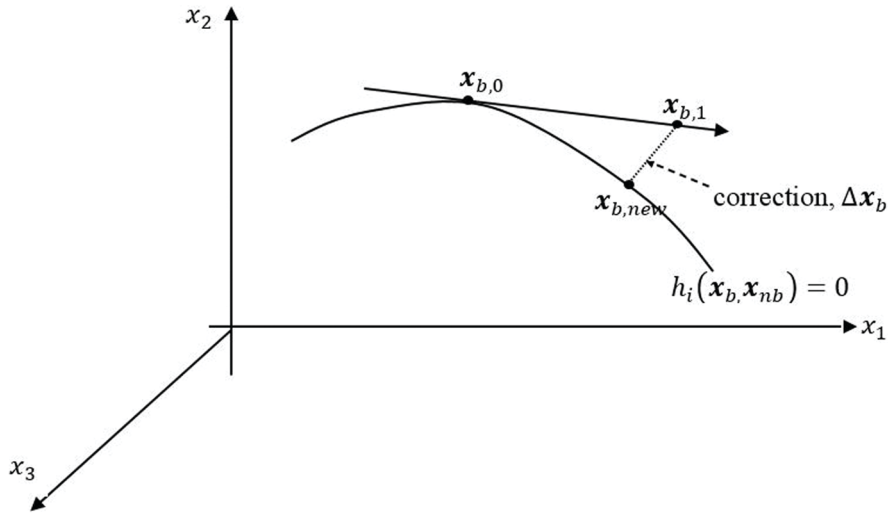


FIGURE 2.24 GRG method, correction (if required) to the basic variables during an iteration by Newton-Raphson method to satisfy $\max h_i(\boldsymbol{x}_1) < \varepsilon, i = 1,2,\ldots$

unconstrained minimum is obtained. With $s = 0.2182$ obtained by line search (see Equation 2.2) – the new $\boldsymbol{x}_1$ is:

$$\boldsymbol{x}_1 = (-0.59, -0.01, 1.03)^T \tag{2.164}$$

The update gives $f(\boldsymbol{x}_1) = 1.52 < f(\boldsymbol{x}_0)$ and is acceptable. Yet, the update needs a check for feasibility before proceeding to the next iteration. The feasibility requirement is to see that $h(\boldsymbol{x}_1)$ is less than, say, $\varepsilon = 10^{-3}$. With $h(\boldsymbol{x}_1) = -2.45$ in this case, $\boldsymbol{x}_1$ is not feasible. Here, a strategy is adopted to make it feasible by keeping $\boldsymbol{x}_{nb}$ fixed whilst suitably varying $x_b = x_3$ so that the feasibility condition $h(x_b, \boldsymbol{x}_{nb}) \cong \varepsilon$ is satisfied (see Figure 2.24). This is equivalent to solving the nonlinear equations $h(x_b, \boldsymbol{x}_{nb}) = 0$ for x_b by a Newton-Raphson method. Starting with $x_{b,0} = 1.03$, the method iteratively yields $x_{b,new}$ as:

$$x_{b,new} = x_{b,new} + \Delta x_b \tag{2.165a}$$

where

$$\Delta x_b = -J^{-1} h(x_b, \boldsymbol{x}_{nb}) \tag{2.165b}$$

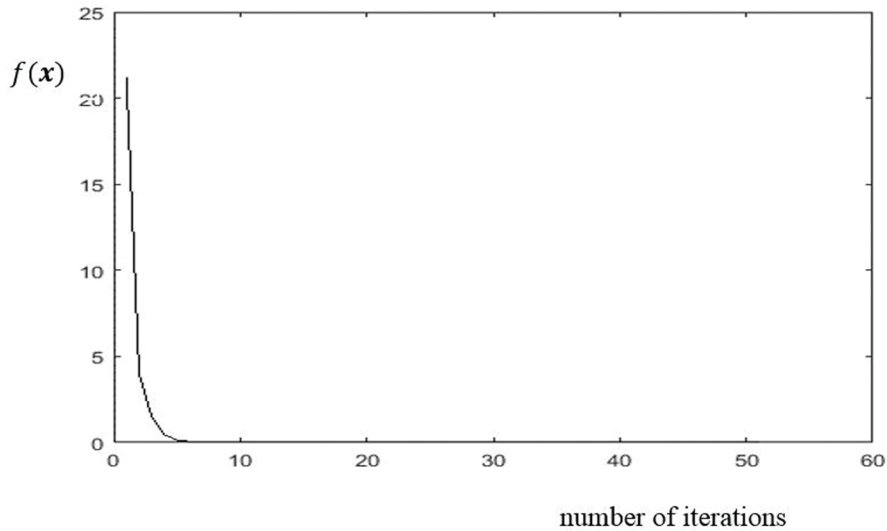


FIGURE 2.25 Minimization of the function in Example 2.7 by GRG method, evolution of objective function with iterations (attaining the minimum value of 1.11E-05 at the end).

$J \in \mathbb{R}^{m \times m}$ is the Jacobian matrix given by the partial gradient of h with respect to x_b. Here $m = 1$. The method converges with $h(\boldsymbol{x}_1) < \varepsilon$ within five iterations. In general, with a greater number of equality constraints, the method is performed until $\max h_i(\boldsymbol{x}_1) < \varepsilon, i = 1,2,\ldots$. In the present problem, with the above correction to x_b and the feasibility condition satisfied, it marks the end of a successful iteration.

Figure 2.25 shows the evolution of the objective function with iterations and its final minimum value $f(\boldsymbol{x}^*) = 0$. The optimum point is $(x_1^*, x_2^*, x_3^*) = (1,1,1)$ whose convergence with iterations is shown in Figure 2.26.

■

In references of Lasdon *et al*. (1973,1978), readers may find a detailed discussion on the development and testing of the GRG method. A comparative study of nonlinear programming methods including the GRG may be found in Floudas and Pardalos (1990) and Yan and Ma (2001).

2.7 METHOD OF FEASIBLE DIRECTIONS

At any iteration, the first step in an optimization problem is to fix a suitable direction $\boldsymbol{d}_k \in \mathbb{R}^n$ at the current point $\boldsymbol{x}_k \in \mathbb{R}^n$. One then proceeds to find the update $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + s_k \boldsymbol{d}_k$ where s_k is the step size. For a minimization problem, d_k needs to be a descent direction in that $(\nabla_x f)^T \boldsymbol{d} \leq 0$ (see

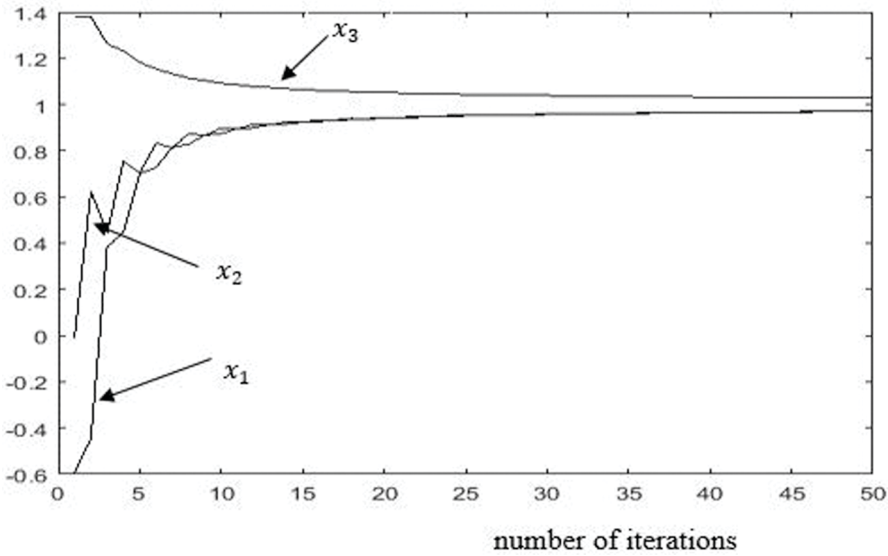


FIGURE 2.26 Minimization of the function in Example 2.7 by GRG method, evolution of design variables x_1, x_2 and x_3 with iterations.

Section 1.6, Chapter 1). For a constrained problem, it is required that $\boldsymbol{d}_k$ be both a descent and a feasible direction. Figure 2.27 shows the cone of such directions $\mathcal{S}:\left\{\boldsymbol{d}_k : \nabla f^T\left(\boldsymbol{x}\right)\boldsymbol{d}_k < 0, \nabla g_1^T\left(\boldsymbol{x}\right)\boldsymbol{d}_k \le 0, \nabla g_2^T\left(\boldsymbol{x}\right)\boldsymbol{d}_k \le 0\right\}$ in a two-dimensional case. In the figure, two inequality constraints are active, i.e. $g_1\left(\boldsymbol{x}\right)=0$ and $g_2\left(\boldsymbol{x}\right)=0$ at $\boldsymbol{x}$ while $g_3\left(\boldsymbol{x}\right)$ is not. Note that the necessary optimality KKT condition in Equation 1.77 (Chapter 1) is satisfied at $\boldsymbol{x}$ if $\mathcal{S}$ is empty, i.e. there exists no vector $\boldsymbol{d}$ at $\boldsymbol{x}$ which is both a descent and a feasible direction. Now, suppose that only inequality constraints are present in the problem. Starting at a feasible point $\boldsymbol{x}_0$ with $\nabla g\left(\boldsymbol{x}_0\right) \le \boldsymbol{0}$, one may arrive, at the end of a line search, at $\boldsymbol{x}_k$ where some p of m inequality constraints are active similar to the case in Figure 2.27.

Define the active set $I=\left\{i : g_i\left(\boldsymbol{x}_k\right)=0, i=1,2,\ldots,p\le m\right\}$. A favorable $\boldsymbol{d}_k$ is obviously the one that a) reduces the objective function and b) keeps the update $\boldsymbol{x}_{k+1}$ within the feasible region with $\nabla g_i^T\left(\boldsymbol{x}\right)\boldsymbol{d}_k \le 0, i \in I$. The method of feasible directions (Zoutendijk 1960) is a direction-finding technique. The method proposes a sub-optimization problem at an iteration k, if one encounters at least one active constraint $g_i\left(\boldsymbol{x}_k\right)=0,\ i\in\left[1,2,..,m\right]$. The formulation introduces a parmeter α as:

$$\alpha = \max\left\{\nabla f^T\left(\boldsymbol{x}_k\right)\boldsymbol{d}_k,\quad \nabla g_i^T\left(\boldsymbol{x}_k\right)\boldsymbol{d}_k, i\in I\right\} \tag{2.166}$$

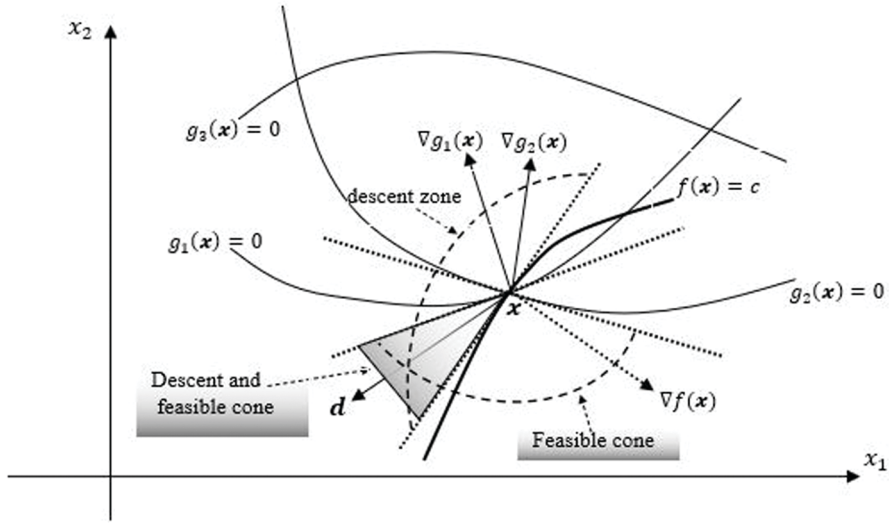


FIGURE 2.27 Constrained optimization, $\boldsymbol{d}$ – a descent and feasible direction, shade region – intersection of descent and feasible cones, $f(\boldsymbol{x}) = c$ is an equipotential curve.

The sub-optimization problem is stated as:

$$\text{Minimize } \alpha$$

$$\text{s. t.}$$

$$\nabla f^T\left(\boldsymbol{x}_k\right)\boldsymbol{d}_k \le \alpha \text{ and}$$

$$\nabla g_i^T\left(\boldsymbol{x}_k\right)\boldsymbol{d}_k \le \alpha, \text{ for each } i \in I$$

$$-1 \le d_{j,k} \le 1, \quad j = 1,2\ldots,n \tag{2.167}$$

The sub-optimization realizes an optimum $\alpha < 0,$ since a negative α ensures by its definition that $\nabla f^T\left(\boldsymbol{x}_k\right)\boldsymbol{d}_k$ and $\nabla g_i^T\left(\boldsymbol{x}_k\right)\boldsymbol{d}_k, i \in I$ are both negative maxima, thereby rendering $\boldsymbol{d}_k$ the most effective descent and feasible direction. The bounds on the elements of $\boldsymbol{d}_k$ in Equation (2.167) lead to the usual normalization of a vector one adopts during an iterative process. Note that this subprogram may fit into an LP problem with the attendant access to the solution techniques associated with LP. Once such a direction $\boldsymbol{d}_k$ is determined by solving the LP problem, a line search is performed to find the largest possible step size s_k followed by an evaluation of the update $\boldsymbol{x}_{k+1}$. The procedure is repeated till convergence is achieved in finding $\boldsymbol{x}^*$. While the method is originally developed by Zoutendijk (1960) for problems with inequality constraints, the method is as well applicable to those with equality constraints (Vanderplaats 1984, Bazaara *et al*. 2006).

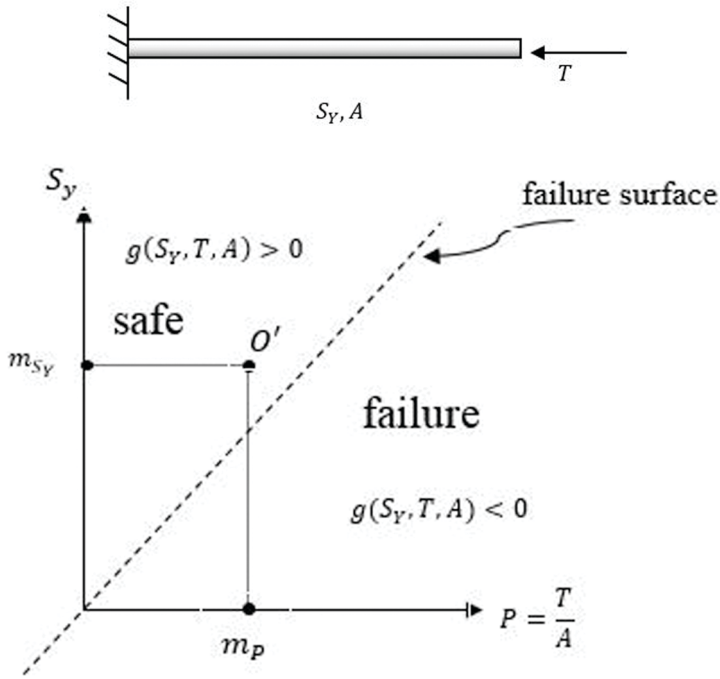


FIGURE 2.28 (a) An axially loaded rod; (b) failure surface $g\left(S_Y, T, A\right) = 0$, load effect $P = \frac{T}{A}$.

Example 2.8. We consider a reliability problem (Ang and Tang 1984, Melchers 2007, Maymon 1998) where it is required to assess the probability of failure of a structure – in this case, an axially loaded rod shown in Figure 2.28. It is given that the material properties and the external load are random variables and have their mean values specified along with their standard deviations. From the description of the problem that follows, we find that the problem of knowing the failure probability may be posed as an optimization problem and we solve it by the method of feasible directions. The design variables are the cross-sectional area A of the rod, the strength variable S_Y in stress units and the axial load T which follow a normal distribution. Table 2.9 gives the details of the three RVs.

Solution. As shown in Figure 2.28, the system under the given load is safe if $S_Y > \frac{T}{A}$. Define $g\left(S_Y, T, A\right) = S_Y - \frac{T}{A}$. In reliability theory, it is customary to call $g\left(S_Y, T, A\right) = 0$ the failure surface which demarcates the design variable space into safe and failure regions (see Figure 2.28a).

TABLE 2.9
Parameters of the Normal Distribution Defining the Three RVs S_y, T and A

RV	Mean	Standard deviation
S_Y	$m_{S_Y} = 2500N/cm^2$	$\sigma_{S_Y} = 75N/cm^2$
T	$m_T = 4166N$	$\sigma_T = 125N$
A	$m_A = 2cm^2$	$\sigma_A = 0.04\,cm^2$

It is straightforward to show that an index β for reliability is given by the shortest distance from the origin to the failure surface in the standard normal space (Figure 2.29); see Appendix B for details. P_f, the probability of failure, is given by $\Phi(-\beta)$ where $\Phi(.)$ is the CDF of a standard normal RV.[§§] The complement of the probability of failure is the reliability given by $(1-P_f)$.

If Z_1, Z_2 and Z_3 are the standard normal RVs corresponding to S_Y, T and A, then:

$$Z_1 = \frac{S_Y - m_{S_Y}}{\sigma_{S_Y}}, \qquad Z_2 = \frac{T - m_T}{\sigma_T},$$

$$Z_3 = \frac{A - m_A}{\sigma_A} \tag{2.168}$$

[§§] Standard normal RV
A normal RV with zero mean and unit variance is known as a standard normal RV. Thus, if Z is a standard normal RV, its *pdf* is given by:

$$\mathbb{f}_Z(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} \qquad \text{i)}$$

The CDF $\mathbb{F}_Z(z)$ of Z is commonly denoted by $\Phi(z)$. Note that a normal RV $X \sim \mathcal{N}(\mu, \sigma)$ may be transformed to a standard normal RV Z by the transformation $Z = (X - \mu)/\sigma$. See Appendix 1 for notes on transformation of RVs.

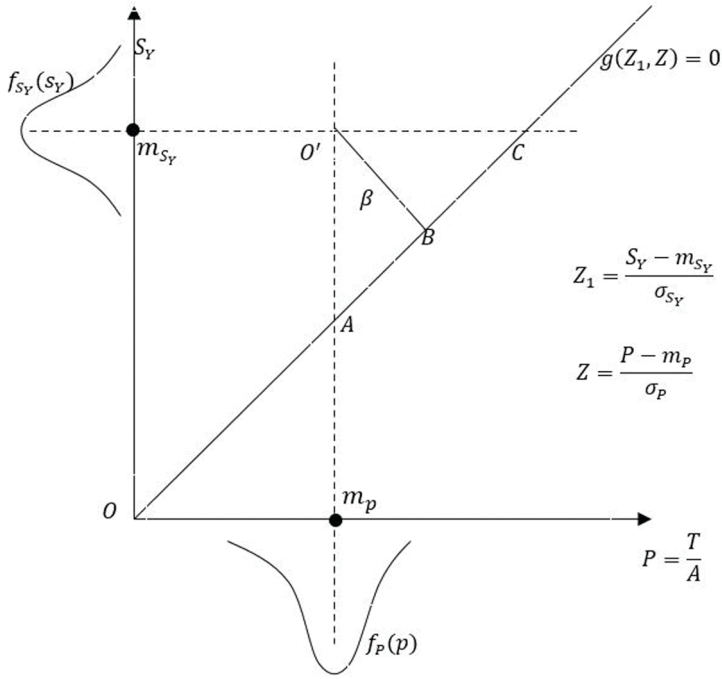


FIGURE 2.29 $f_{S_Y}(s_Y)$, $f_P(p)$–normal *pdf*s of S_Y and the load effect P respectively; failure surface $g(Z_1, Z) = 0$ where Z_1 and Z are standard normals of S_Y and P, respectively.

The failure surface in terms of the standard normals may be written as:

$$g(Z_1, Z_2, Z_3) = \left(\sigma_{S_Y} Z_1 + m_{S_Y}\right)\left(\sigma_A Z_3 + m_A\right) - \left(\sigma_T Z_2 + m_T\right) = 0 \qquad (2.169)$$

Now, to find the reliability index $\beta = \sqrt{Z_1^2 + Z_2^2 + Z_3^2}$, we formulate the following optimization problem:

$$\text{minimize } f(Z_1, Z_2, Z_3) = Z_1^2 + Z_2^2 + Z_3^2$$

s. t.

$$\left(\sigma_{S_Y} Z_1 + m_{S_Y}\right)\left(\sigma_A Z_3 + m_A\right) - \left(\sigma_T Z_2 + m_T\right) \le 0 \qquad (2.170)$$

The optimization problem, being of a relatively low dimension, may be conveniently solved by, say, the Lagrange multipliers method (Section 1.6, Chapter 1). However, we use this example to illustrate the application of the method of feasible directions.

Start of iterations

Let $\boldsymbol{x}_0 = \left(S_Y, T, A\right)^T = \left(2000, 4200, 2.1\right)$ with corresponding $\boldsymbol{z}_0 = (Z_1,\ Z_2,\ Z_3)^T = (-6.67, 0.272, 2.5)^T$. Also, $f\left(\boldsymbol{z}_0\right) = 50.81$. Since $\boldsymbol{x}_0$ lies on the failure surface, the constraint $g\left(Z_1, Z_2, Z_3\right)$ is active. We now proceed to find the descent and feasible direction $\boldsymbol{d}_0$. The sub-optimization problem (Equation 2.167) is formulated as:

$$\text{minimize } \alpha$$

s. t.

$$\nabla f^T\left(\boldsymbol{z}_0\right)\boldsymbol{d}_0 \le \alpha \text{ and}$$

$$\nabla g^T\left(\boldsymbol{z}_0\right)\boldsymbol{d}_0 \le \alpha$$

$$-1 \le d_{j,0} \le 1, \quad j = 1,2,3 \tag{2.171}$$

Here $\nabla f\left(\boldsymbol{z}\right) = \left(2Z_1, 2Z_2, 2Z_3\right)^T$ and $\nabla g\left(\boldsymbol{z}\right) = \{\ \sigma_{S_Y}\left(\sigma_A Z_3 + m_A\right),\ -\sigma_T,\ \sigma_A\left(\sigma_{S_Y} Z_1 + m_{S_Y}\right)\}^T$. Substituting $d_i = s_i - 1$ to keep the components of the search direction non-negative, the problem in Equation (2.171) in an explicit form is:

$$\text{minimize } \alpha$$

s. t.

$$2Z_1 s_1 + 2Z_2 s_2 + 2Z_3 s_3 - \alpha - 2\left(Z_1 + Z_2 + Z_3\right) + y_1 = 0$$

$$\sigma_{S_Y}\left(\sigma_A Z_3 + m_A\right)s_1 - \sigma_T s_2 + \sigma_A\left(\sigma_{S_Y} Z_1 + m_{S_Y}\right)s_3 - \alpha$$

$$-\left\{\sigma_{S_Y}\left(\sigma_A Z_3 + m_A\right) - \sigma_T + \sigma_A\left(\sigma_{S_Y} Z_1 + m_{S_Y}\right)\right\} + y_2 = 0$$

$$s_1 + y_3 = 2$$

$$s_2 + y_4 = 2$$

$$s_3 + y_5 = 2$$

$$s_i \ge 0 \tag{2.172}$$

TABLE 2.10
First Iteration and the Initial Simplex Tableau for Solving the LP Problem in Example 2.8

Basic variables	Original variables			Slack variables					b_i (RHS elements)
	S_1	S_2	S_3	y_1	y_2	y_3	y_4	y_5	
y_2	170.84	–125.54	75	–1	1	0	0	0	108.6
y_3	1	0	0	0	0	1	0	0	2
y_4	0	1	0	0	0	0	1	0	2
y_5	0	0	1	0	0	0	0	1	2
α	–13.34	0.544	5	1	0	0	0	0	–3.9

The last Equation (2.172) represents an LP problem. $y_i, i = 1,2,3,4,5$ are slack variables. Thus, one may obtain a solution by combining the method of feasible directions with the simplex method. To this end, we proceed as follows.
From the first constraint in (2.172), α may be expressed as:

$$\alpha = 2Z_1 s_1 + 2Z_2 s_2 + 2Z_3 s_3 - 2\left(Z_1 + Z_2 + Z_3\right) + y_1 \tag{2.173}$$

Also, the first two constraints could be merged into one by eliminating α. With this simplification, the simplex Tableu in Table 2.10 shows the initial set-up to solve the LP problem by the simplex method (see Section 2.5).

Solving the above LP problem, one obtains $s_1 = 2$, $s_2 = 1.856$, $s_3 = 0$ and $\alpha = -25.67$. This result gives the descent and feasible direction for the original problem in (2.170) as $\boldsymbol{d}_0 = \left(1, 0.856, -1\right)^T$. A line search with this direction gives the step size s_0 as 3.27. The update $\boldsymbol{z}_1$ is obtained as $\boldsymbol{z}_0 + s_0 \boldsymbol{d}_0 = \left(-3.4, 3.07, -0.77\right)^T$. This corresponds to $\boldsymbol{x}_1 = \left(S_Y, T, A\right)_1^T = \left(2245, 4550, 1.97\right)$. The point $\boldsymbol{z}_1$ is away from the failure surface with $g\left(S_Y, T, A\right) = S_Y - T/A = 64.65 \neq 0$. Here $\boldsymbol{x}_1$ is corrected so as to satisfy the constraint surface before the next iteration begins with the direction-finding step. This is realized by keeping Z_2 and Z_3 fixed and adjusting Z_1 so as to satisfy $g\left(Z_1, Z_2, Z_3\right) = 0$ which yields $\boldsymbol{z}_1 = \left(2.522, 3.07, -0.77\right)^T$.

Second iteration

At this stage, $f\left(\boldsymbol{z}_1\right) = 16.38$ and so the distance from O' to the failure surface is $\beta = \left\|\boldsymbol{z}_1\right\| = \sqrt{Z_1^2 + Z_2^2 + Z_3^2} = 4.047$. As before, we again formulate the sub-optimization problem of minimizing α as defined in Equation (2.164). Table 2.11 shows the simplex Tableu detailing the initial set-up for this iteration.

The solution to the LP problem is $s_1 = 0.7508$, $s_2 = 0$, $s_3 = 0$ and $\alpha = -3.79$. Hence, the descent and feasible direction for the original problem in (2.170) is

TABLE 2.11
Second Iteration and the Initial Simplex Tableau for Solving the LP Problem in Example 2.8

Basic variables	Original variables			Slack variables					b_i (RHS elements)
	S_1	S_2	S_3	y_1	y_2	y_3	y_4	y_5	
y_2	152.73	–131.14	93.97	–1	1	0	0	0	114.68
y_3	1	0	0	0	0	1	0	0	2
y_4	0	1	0	0	0	0	1	0	2
y_5	0	0	1	0	0	0	0	1	2
α	–5.04	6.14	–1.54	1	0	0	0	0	–0.444

TABLE 2.12
Third Iteration and the Initial Simplex Tableau for Solving the LP Problem in Example 2.8

Basic variables	Original variables			Slack variables					b_i (RHS elements)
	S_1	S_2	S_3	y_1	y_2	y_3	y_4	y_5	
y_2	150.71	–129.52	94.99	–1	1	0	0	0	107.74
y_3	1	0	0	0	0	1	0	0	2
y_4	0	1	0	0	0	0	1	0	2
y_5	0	0	1	0	0	0	0	1	2
α	–5.45	4.52	–3.16	1	0	0	0	0	–4.09

$\boldsymbol{d}_1 = (-0.2492, -1, -1)^T$. A line search with this direction yields $s_1 = 0.8106$. This is followed by evaluation of the update $z_2 = (-2.724, 2.259, -1.581)^T$ which corresponds to $\boldsymbol{x}_2 = (S_Y, T, A)_1^T = (2296, 4448, 1.936)^T$. At the end of the second iteration, $f(z_2) = 15.02$ and the distance of O' to the failure surface at z_2 is 3.876 (Figure 2.29). The point $\boldsymbol{x}_2$ lies almost on the failure surface so that the constraint $g(S_Y, T, A)$ remains active.

Third iteration

The simplex tableau at the beginning of this iteration is given in the Table 2.12.

Solving the LP problem gives $s_1 = 2$, $s_2 = 1.4954$, $s_3 = 0$ and $\alpha = -4.13$ so that $\boldsymbol{d}_2 = (1, 0.4954, -1)^T$. A line search gives $s_2 = 0.011$. The update $z_3 = (-2.713, 2.265, -1.592)^T$. The corresponding update $\boldsymbol{x}_3 = (S_Y, T, A)^T = (2296.53, 4449.13, 1.936)^T$. $f(z_3)$ is 15.025 and the distance of O' (Figure 2.29) to the failure surface at z_3 is 3.8762.

Comparison with the result of the previous iteration indicates that convergence is achieved and the reliability index β (Figure 2.29) may be taken as 3.8762. The probability of failure P_f is $\Phi(-\beta) = 5.305\text{E-}5$ (from a standard normal distribution table – for example in Ang and Tang [1975]. ■

Solution *to Example 2.8 by Lagrange multipliers method:*
Suppose that the solution to the optimization problem in Equation (2.170) is attempted by the Lagrange multiplier method. In this case, the constrained minimization problem is transformed to an unconstrained one as:

$$L(z,l) = f(z) + \lambda g(z) \tag{2.174}$$

where λ is the Lagrange multiplier. Then setting the gradients (derivatives with respect to $\mathbf{z}$ and λ) to zero, one gets the KKT optimality conditions:

$$\frac{\partial L}{\partial Z_1} = 0,\ \frac{\partial L}{\partial Z_2} = 0,\ \frac{\partial L}{\partial Z_3} = 0 \text{ and}$$

$$\frac{\partial L}{\partial \lambda} = 0 \tag{2.175}$$

If explicitly expressed in terms of the given parameters, the optimality conditions take the form of coupled (and nonlinear) algebraic equations:

$$2Z_1 - \lambda\sigma_{Sy} = 0$$

$$2Z_2\left(\sigma_A Z_3 + m_A\right) + \lambda\sigma_T = 0$$

$$2Z_3\left(\sigma_A Z_3 + m_A\right)^2 - \lambda\left(\sigma_T Z_2 + m_T\right)\sigma_A = 0$$

$$\left(\sigma_{Sy} Z_1 + m_{Sy}\right) * \left(\sigma_A Z_3 + m_A\right) - \left(\sigma_T Z_2 + m_T\right) = 0 \tag{2.176a-d}$$

Let the initial vector $\boldsymbol{x}_0$ be $\left(S_Y, T, A\right)^T = \left(2000, 4200, 2.1\right)^T$. The corresponding vector $\boldsymbol{z}_0$ in the standard normal space is $\left(Z_1, Z_2, Z_3\right)^T = \left(-6.67, 0.272, 2.5\right)^T$. The nonlinear algebraic equations in (2.176) are solved by the Newton-Raphson method. Convergence is achieved in five iterations. The optimal solution is $\left(Z_1^*, Z_2^*, Z_3^*\right)^T = \left(-2.637, 2.674, -1.675\right)^T$ and the corresponding

vector $(S_Y, T, A) = (2302.23, 4450.2, 1.933)$. The solution for the reliability index β is $\|z_1\| = \sqrt{Z_1^2 + Z_2^2 + Z_3^2} = 3.8638$ and the probability of failure is $P_f = \phi(-\beta) = \phi(-3.8638) = 5.58\text{E}-5$. The result is close to the one obtained by the method of feasible directions. ■

Solution *to Example 2.8 by MC simulation:*
For this example, given the expression for the failure surface $g(S_Y, T, A) = S_Y - T/A$, P_f is obtainable by direct simulation of the RVs (see Appendix 1 for simulation of RVs). Given the probability distributions of the RVs, the realizations of the three normal RVs S_Y, T and A are numerically obtained. The relative frequency of occurrences of failure events $S_Y - T/A < 0$ is thus computed. As the number of realizations increases, the relative frequency should approach the P_f. One such simulation with one hundred thousand realizations of the RVs is shown in Figure 2.30 and P_f is obtained as $5E-05$. Here, the generation of realizations for normal RVs warrants a special mention. In Chapter 1, MC simulation of a uniformly distributed

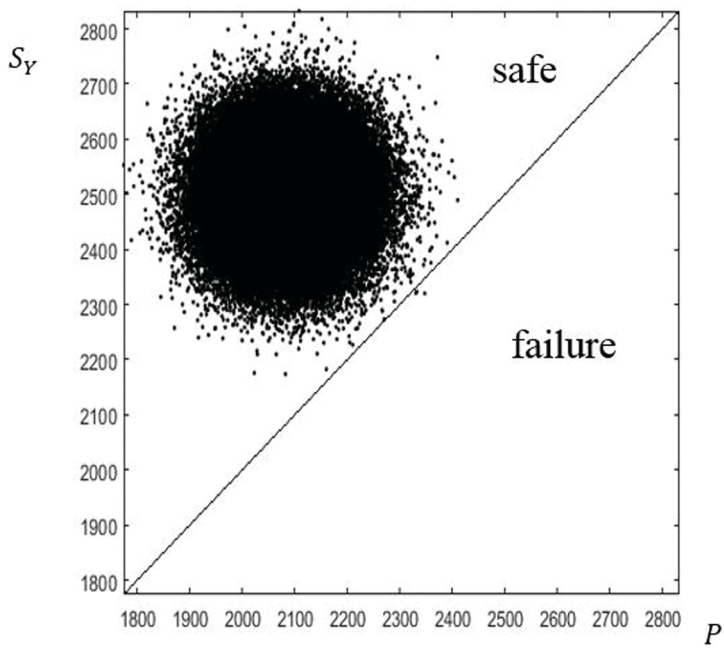


FIGURE 2.30 Solution to Example 2.8 by MC simulation; number of simulations = 1E05, probability of failure $P_f = 5\text{E}-05$.

(UD) RV $X \approx U(0-1)$ is described. RVs of other probability distributions may be obtained from X by an appropriate transformation. Box-Muller transformation is utilized in this example to simulate the three normal RVs. See Appendix 1 for a description of the transformation.

■

A solution by the method of feasible directions (Topkis and Veinott 1967, Ravindran *et al*. 2007) is known to be susceptible to oscillations due to sudden changes introduced in the search direction. This in turn may prevent the method from converging. One remedy may be to have I as a set of near-active constraints (instead of a strictly active set) defined as: $I : \left\{ i : g_i(\boldsymbol{x}_k) + \varepsilon \geq 0, i = 1,2,\ldots, p \leq m \right\}$ for some small $\varepsilon > 0$ and proceed with the iterative steps described earlier. Topkis and Veinott0 (1967) in fact modified the method in order to ensure convergence by involving both the active and inactive constraints in the direction-finding step. With the modification incorporated into the step, the sub-optimization problem in (2.167) is restated as:

minimize α

s. t.

$$\nabla f^T(\boldsymbol{x}_k)\boldsymbol{d}_k \leq \alpha \text{ and}$$

$$\nabla g_i^T(\boldsymbol{x}_k)\boldsymbol{d}_k \leq \alpha + g_i(\boldsymbol{x}_k), \text{ for each } i \in m$$

$$-1 \leq d_{j,k} \leq 1, \quad j = 1,2\ldots,n \tag{2.177}$$

Adding $g_i(\boldsymbol{x}_k)$ to each constraint prevents sudden changes in the search direction and assures convergence of the method.

2.8 METHOD OF GRADIENT PROJECTION

Similar to the method of feasible directions, the gradient projection is also a direction-finding technique. The method was proposed by Rosen (1960) initially for problems with linear constraints and later extended to nonlinear problems (Rosen 1961). In either case, the method, as its name indicates, employs a projection matrix to operate on the negative gradient $-\nabla f(\boldsymbol{x})$ for obtaining $\boldsymbol{d}$ which is both a descent and a feasible direction. A matrix $\boldsymbol{P}$ is called a projection matrix if $\boldsymbol{P}^T = \boldsymbol{P}$ (symmetry) and $\boldsymbol{PP} = \boldsymbol{P}$. For an identity matrix $\boldsymbol{I}$ of the same size as $\boldsymbol{P}$, $\boldsymbol{I} - \boldsymbol{P}$ is also a projection matrix. For example, let $h(\boldsymbol{x}) = \boldsymbol{Cx} - \boldsymbol{D} = \boldsymbol{0} \in \mathbb{R}^I$ be a

given set of l active equality constraints with $\boldsymbol{C} \in \mathbb{R}^{l \times n}$ and $\boldsymbol{D} \in \mathbb{R}^{l}$. Also let $\boldsymbol{Q} = \left[\nabla h_i(\boldsymbol{x}), i = 1,2,\ldots,l\right] \in \mathbb{R}^{n \times l}$ with each $\nabla h_i(\boldsymbol{x})$ being an n-dimesional vector. Then note that $\boldsymbol{P} = \left[\boldsymbol{I} - \boldsymbol{Q}\left[\boldsymbol{Q}^T\boldsymbol{Q}\right]^{-1}\boldsymbol{Q}^T\right] \in \mathbb{R}^{n \times n}$ is a projection matrix. With x chosen to satisfy the equality constraints and thus being a feasible point, the gradient projection method claims that $\boldsymbol{d} = -\boldsymbol{P}\nabla f(\boldsymbol{x})$ is a descent direction, which is also a feasible direction.

Proof: We formulate the following sub-optimization problem to derive d :

minimize $\mathbb{f}(\boldsymbol{d}) = \boldsymbol{d}^T \nabla f(\boldsymbol{x})$

s. t.

$$\boldsymbol{Q}^T\boldsymbol{d} = 0 \text{ and}$$

$$\boldsymbol{d}^T\boldsymbol{d} = 1 \tag{2.178}$$

Using Lagrange multipliers, we restate the problem as an unconstrained one as:

$$\text{minimize } L = \boldsymbol{d}^T \nabla f(\boldsymbol{x}) + \lambda^T \boldsymbol{Q}^T\boldsymbol{d} + \mu\left(\boldsymbol{d}^T\boldsymbol{d} - 1\right) \tag{2.179}$$

$\boldsymbol{\lambda} \in \mathbb{R}^{l}$ and $\mu \in \mathbb{R}$ are the Lagrange multipliers. The optimality conditions are:

$$\frac{\partial L}{\partial \boldsymbol{d}} = \boldsymbol{0} \Rightarrow \nabla f(\boldsymbol{x}) + \boldsymbol{Q}\lambda + 2\mu\boldsymbol{d} = 0$$

$$\frac{\partial L}{\partial \boldsymbol{\lambda}} = 0 \Rightarrow \boldsymbol{Q}^T\boldsymbol{d} = 0$$

$$\frac{\partial L}{\partial \mu} = 0 \Rightarrow \boldsymbol{d}^T\boldsymbol{d} - 1 = 0 \tag{2.180a,b,c}$$

From these conditions, we obtain:

$$\boldsymbol{d} = \frac{1}{2\mu}\left(\nabla f(\boldsymbol{x}) + \boldsymbol{Q}\boldsymbol{\lambda}\right)\left(\text{from Equation 2.180a}\right) \tag{2.181a}$$

$$\Rightarrow \boldsymbol{Q}^T\boldsymbol{d} = \boldsymbol{Q}^T\left(\nabla f(\boldsymbol{x}) + \boldsymbol{Q}\boldsymbol{\lambda}\right) = \boldsymbol{0}\left(\text{from Equation 2.180b and since } \mu \neq 0\right)$$

$$\Rightarrow \boldsymbol{\lambda} = -\left[\boldsymbol{Q}^T\boldsymbol{Q}\right]^{-1}\boldsymbol{Q}^T\nabla f(\boldsymbol{x}) \tag{2.181b}$$

Substituting $\boldsymbol{\lambda}$ in Equation (2.181a) gives:

$$\boldsymbol{d} = -\left[\boldsymbol{I} - \boldsymbol{Q}\left[\boldsymbol{Q}^T\boldsymbol{Q}\right]^{-1}\boldsymbol{Q}^T\right]\nabla f(\boldsymbol{x})$$

$$= -\boldsymbol{P}\nabla f(\boldsymbol{x}) \tag{2.182}$$

Now, we have:

$$\nabla f(\boldsymbol{x})^{\mathrm{T}}\,\boldsymbol{d} = -\nabla f(\boldsymbol{x})^T\,\boldsymbol{P}\nabla f(\boldsymbol{x})$$

$$= -\nabla f(\boldsymbol{x})^T\,\boldsymbol{P}\boldsymbol{P}\nabla f(\boldsymbol{x})$$

$$= -\nabla f(\boldsymbol{x})^T\,\boldsymbol{P}^T\boldsymbol{P}\nabla f(\boldsymbol{x}) = -\boldsymbol{P}\nabla f(\boldsymbol{x})^2 < 0 \tag{2.183}$$

Hence $\boldsymbol{d} = -\boldsymbol{P}\nabla f(\boldsymbol{x})$ is a descent direction. Further, $\boldsymbol{d}$ satisfies $\boldsymbol{Q}^T\boldsymbol{d} = \boldsymbol{0}$ (Equation 2.180b). Hence if $d \neq 0$, P projects the gradient of the objective function onto the null space of $\boldsymbol{Q}$. This is equivalent to:

$$\left[\nabla h_1(\boldsymbol{x})^T\ \nabla h_2(\boldsymbol{x})^T \quad \ldots \quad \nabla h_l(\boldsymbol{x})^T\right]\boldsymbol{d} = 0$$

$$\Rightarrow \nabla h_1(\boldsymbol{x})^T\,\boldsymbol{d} = 0, \quad \nabla h_2(\boldsymbol{x})^T\,\boldsymbol{d} = 0, \quad \ldots \quad \nabla h_l(\boldsymbol{x})^T\,\boldsymbol{d} = 0 \tag{2.184}$$

which ensures that $\boldsymbol{d}$ is also a feasible direction. The proof is complete. ♦

Geometric interpretation of the projection matrix $\boldsymbol{P}$

In the case of unconstrained optimization, the entire hyperspace in $\mathbb{R}^n$ is a feasible region and one may proceed with the steepest descent direction $-\nabla f(\boldsymbol{x})$ and try for an improved feasible solution. But in the presence of constraints, $-\nabla f(\boldsymbol{x})$ may no longer be the feasible direction. The gradient projection method uses the matrix $\boldsymbol{P}$ to project $-\nabla f(\boldsymbol{x})$ on to the binding constraints so as to move along $\boldsymbol{d}$, the descent and feasible direction (Figure 2.31).

Suppose that inequality constraints are also present in the form $g(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{B} \leq \boldsymbol{0} \in \mathbb{R}^m$. Let p out of m constraints be active such that $g_j(\boldsymbol{x}) = 0, j = j_1, j_2, \ldots, j_p$ with $j_i \in [1,2,\ldots,m]$. Combined with the equality constraints $h_i(\boldsymbol{x}), i = 1,2,\ldots,l$, we now have $\boldsymbol{Q} = \left[\nabla h_i(\boldsymbol{x}), i = 1,2,\ldots,l,\ \nabla g_j(\boldsymbol{x}),\right.$

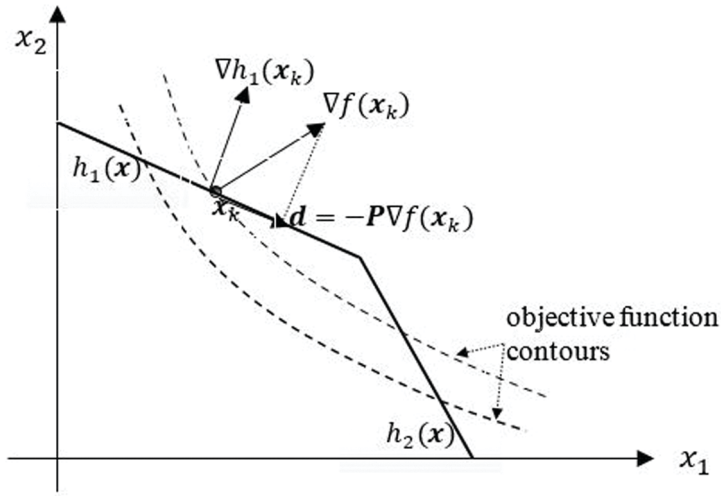


FIGURE 2.31 Method of gradient projection: equality constraints: $h_1(\boldsymbol{x})$ and $h_2(\boldsymbol{x})$ with $h_1(\boldsymbol{x})$ being the binding constraint at $\boldsymbol{x}_k$, descent and feasible direction $\boldsymbol{d} = -\boldsymbol{P}\nabla f(\boldsymbol{x})$ so that $\nabla h_1(\boldsymbol{x})^T \boldsymbol{d} = 0$.

$j = j_1, j_2, \ldots, j_p \Big] \in \mathbb{R}^{n\times(l+p)}$. Moreover, $\boldsymbol{P} = \Big[\boldsymbol{I} - \boldsymbol{Q}\big[\boldsymbol{Q}^T\boldsymbol{Q}\big]^{-1}\boldsymbol{Q}^T\Big] \in \mathbb{R}^{n\times n}$ and $\boldsymbol{d} = -\boldsymbol{P}\nabla f(\boldsymbol{x})$.

Nonlinear constraints

The method may be applied to problems with nonlinear constraints of equality or inequality types (Rosen 1961, Haug and Arora 1979). In this case, the projected gradient produces a direction $\boldsymbol{d}$ that is tangential to the nonlinear constraint surface. Hence, it is required to provide a correction to the new point $\boldsymbol{x}_{k+1}$ so that it is brought back to the constraint surface and becomes a feasible point. The correction strategy is used in the earlier methods also – see Figure 2.24 for the GRG method. The following example illustrates the method with a nonlinear constraint.

Example 2.9. We apply the gradient projection method to the reliability problem in Example 2.8.

Solution. The optimization problem as defined in Equation (2.170) has $n = 3$ and $l = 1$. The gradients $\nabla f(\boldsymbol{z}) = \left(2Z_1, 2Z_2, 2Z_3\right)^T$. The constraint surface is given by $g(\boldsymbol{z}) = \left(\sigma_{S_Y} Z_1 + m_{S_Y}\right)\left(\sigma_A Z_3 + m_A\right) - \left(\sigma_T Z_2 + m_T\right) = 0 = h(\boldsymbol{z})$ and

$$\nabla h(\boldsymbol{z}) = \left\{\sigma_{S_Y}\left(\sigma_A Z_3 + m_A\right), -\sigma_T,\ \sigma_A\left(\sigma_{S_Y} Z_1 + m_{S_Y}\right)\right\}^T.$$

1st iteration

Let us start with a feasible point $z_0 = \left(Z_1, Z_2, Z_3\right)^T = \left(-6.67, 0.272, 2.5\right)^T$ corresponding to a choice of $\left(S_Y, T, A\right)^T = \left(2000, 4200, 2.1\right)$. One has $\boldsymbol{Q} = \nabla h(z)$ and $\boldsymbol{P}$ is given by:

$$\boldsymbol{P} = \left[\boldsymbol{I} - \boldsymbol{Q}\left[\boldsymbol{Q}^T\boldsymbol{Q}\right]^{-1}\boldsymbol{Q}^T\right]$$

$$= [0.4703 \quad 0.4204 \quad -0.2690$$

$$0.4204 \quad 0.6663 \quad 0.2135$$

$$-0.2690 \quad 0.2135 \quad 0.8634] \qquad (2.185)$$

$\boldsymbol{I}$ is a 3×3 identity matrix. $\boldsymbol{P}$ is also a symmetric 3×3 matrix with $\boldsymbol{PP} = \boldsymbol{P}$ by definition. The normalized $\boldsymbol{d} = -\boldsymbol{P}\nabla f(z)$ is obtained as $\boldsymbol{d} = \left(0.633,\ 0.358,\ -0.687\right)^T$. The direction is orthogonal to $\nabla h(z)$ as is evident from $\nabla h(z)\boldsymbol{d} = -3.5527E\text{-}14 \cong 0$ (similar to Equation 2.184) and hence the projected gradient lies on the hyperplane tangential at z_0 to the failure surface $g(z) = 0$. Proceeding in this direction, we get the step size by a line search as $s = 5.84$ and the update $z_1 = z_0 + s_1\boldsymbol{d}_0 = \left(-2.975,\ 2.361, -1.511\right)^T$ which corresponds to $\left(S_Y, T, A\right)^T = \left(2276.9,\ 4461.1,\ 2.1\right)$. The update is away from the nonlinear failure surface as expected. A correction is applied as in Figure 2.24 and the revised update is $z_1 = \left(-2.666, 2.361, -1.511\right)^T$. The distance of O' (Figure 2.29) to the failure surface at z_1 is $\sqrt{Z_1^2 + Z_2^2 + Z_3^2} = 3.8683$. At the end of just the first iteration, the distance is fairly close to the final value of β obtained by the method of feasible directions.

Second and third iterations

Repeating the above steps, we obtain the updates z_2 and z_3 as $(-2.642,\ 2.271,\ -1.671)^T$ and $\left(-2.637,\ 2.274, -1.675\right)^T$ respectively. Corresponding values for the distance of O' (Figure 2.29) to the failure surface are almost identical, which is 3.8638. With this value accepted as the reliability index β, the probability of failure is 5.58E-5. The final point on the failure surface corresponding to z_3 is $\left(S_Y, T, A\right)^T = \left(2302.2,\ 4450.2,\ 1.933\right)$ ■

Additional features of the gradient projection method

During the iterative process, it is possible that $\boldsymbol{d} = \boldsymbol{0}$, i.e., $-\boldsymbol{P}\nabla f(\boldsymbol{x}) = \boldsymbol{0}$. Since $\boldsymbol{d} = -\left[\boldsymbol{I} - \boldsymbol{Q}\left[\boldsymbol{Q}^T\boldsymbol{Q}\right]^{-1}\boldsymbol{Q}^T\right]\nabla f(\boldsymbol{x}) = \boldsymbol{0}$ and $\boldsymbol{\lambda} = -\left[\boldsymbol{Q}^T\boldsymbol{Q}\right]^{-1}\boldsymbol{Q}^T\nabla f(\boldsymbol{x})$ (see Equations 2.181b and 2.182), one has:

$$\boldsymbol{d} = \boldsymbol{0} \Rightarrow \nabla f(\boldsymbol{x}) = \boldsymbol{Q}\boldsymbol{\lambda}$$

$$\Rightarrow -\nabla f(\boldsymbol{x}) = \sum_{i=1}^{l}\lambda_i \nabla h_i(\boldsymbol{x}) + \sum_{j=1}^{j_p}\lambda_{l+j}\nabla g_j(\boldsymbol{x}) \tag{2.186}$$

If all $\lambda_{l+j}, j = 1,2,..,j_p$ are non-negative, the KKT conditions for optimality are satisfied and the iterative process is terminated. In case any of the multipliers λ_{l+j} is negative, it means that the corresponding $\nabla g_j(\boldsymbol{x})$ makes an obtuse angle with $-\nabla f(\boldsymbol{x})$ and the KKT condition is violated (Figure 1.25 in Chapter 1). Hence the gradient $\nabla g_j(\boldsymbol{x})$ is discarded from the $\boldsymbol{Q}$ matrix and not considered in the next iteration. Readers may find further insights on the method in Fox (1982) and Iusem (2003) regarding its convergence issues and applications to different test problems.

CONCLUDING REMARKS

The subject of optimization has a long history, often with deep intellectual underpinnings. We have presented only a brief overview of the early developments which are particularly gradient-based. While no attempt is made in presenting these methods in any sequential order, emphasis is placed more on highlighting the innovative concepts underlying these methods. Methods that are suited for unconstrained optimization problems are first described, followed by a few of those that involve constraints. All real-world optimization problems are generally constrained. Also, constrained optimization methods are invariably iterative in nature. These methods, in general, transform the original problem to an unconstrained one over each iteration. This obviously signifies the role of unconstrained optimization methods in solving a problem with constraints. Among these methods, the CG and quasi-Newton methods such as DFP and BFGS which are based on the concept of conjugacy, stand out as powerful algorithms. The CG method merits a special mention. The technique originally developed to solve large scale linear systems of equations $AX = b$ was effectively made use of in the CG method for non-linear optimization problems. Similarly, the development of LP solution methods also ranks high in the echelon of schemes for constrained optimization problems. These LP methods may have been originally meant for only solving problems with linear objective functions and constraints. Nevertheless, they do find an echo in several constrained

optimization techniques such as sequential linear programming and the method of feasible directions.

These traditional derivative-based methods have doubtless provided an orientation for subsequent research effort leading to a large assortment of numerical optimization techniques. Some of these efforts culminated in the emergence of derivative-free methods. These include the evolutionary / stochastic search methods that are sometimes (or, perhaps oftentimes) motivated by sociological or biological phenomena. It is this last class of methods that we deal with in Chapter 3.

NOTATIONS

A	area of cross section an axially loaded rod in Example 2.8
A_i	areas of cross-section of members in the plane truss in Figure 2.12
A	a matrix
b	a column vector (Equation 2.10)
B and C	matrices in Equation (2.156a) and Table 2.8
c	a real constant
$d_i, i = 0,1,\ldots,n-1Q$	-conjugate directions
$f(x)$	scalar objective function in $\boldsymbol{x} \in \mathbb{R}^n$
$f(Z_1, Z_2, Z_3)$	objective function in terms of standard normals (Example 2.8)
$\hat{f}(r,x)$	augmented objective function (Equation 2.89)
$\hat{f}(\mu, r_k, x)$	augmented objective function (Equation 2.117)
$\hat{f}(\lambda, r_k, x, y)$	augmented objective function (Equations 2.120 and 2.127)
F^e	elemental heat source vector in Example 2.1
F	– heat source vector in the FE model (Example 2.1)
–	force vector in the FE model (Example 2.2)
$\mathbb{F}_k$	approximation to an inverse of Hessian matrix $\left(\approx H_k^{-1}\right)$, at the k^{th} iteration (in DFP method)
g_k	gradient vector at the k^{th} iteration
$g_j(x),\ j = 1,2,\ldots$	inequality constraints
$\boldsymbol{g}_j(x), j = 1,2,\ldots$	linear inequality constraints in the SQP method
$g(S_Y, T, A)$	failure surface (Example 2.8)
$g(Z_1, Z_2, Z_3)$	failure surface in terms of standard normals (Example 2.8)
$\hbar_i(\boldsymbol{x}),\ i = 1,2,\ldots$	equality constraints
$h_i(x),\ i = 1,2,\ldots$	linear equality constraints in the SQP method
$H = \nabla^2 f(x)$	the Hessian matrix
H_k	approximation to the Hessian matrix in the BFGS method

$I\left\{i : g_i\left(x_k\right) = 0,\ i = 1,2,\ldots, p \le m\right\}$	an active set of constraints in the method of feasible directions
I	identity matrix
J	Jacobian matrix in Example 2.7
k_x and k_y	thermal conductivities of a material in x and y directions
$K^e \quad 3\times3$	element matrix in Example 2.1
K	assembled matrix in the FE model (Examples 2.1 and 2.2)
L	length parameter in Figure 2.12
$L\left(x,\lambda,\mu\right)$	Lagrangian in Equation (2.131)
$L\left(x,\lambda,\mu,r\right)$	Lagrangian in Equation (2.132) including a penalty parameter
$\mathrm{L}\left(\mathrm{z},\lambda\right)$	Lagrangian in Equation (2.174)
L	lower triangular matrix obtained through Cholesky decomposition of a matrix
L_k Decomposition of $\mathbb{F}_k$	lower triangular matrix obtained through Cholesky in the DFP method
m_A	mean value of the RV A (Example 2.8)
m_{S_y}	mean value of the RV S_Y (Example 2.8)
m_T	mean value of the RV T (Example 2.8)
M	positive real constant
M	preconditioning matrix in the CG method (Table 2.2)
$\frac{M}{m}$	condition number
$p_k = x_{k+1} - x_k$	(in the DFP method)
P_f	probability of failure
P	projection matrix in the method of gradient projection (section 2.8)
$Q\left(x,y\right)$	heat source in the Example 2.1
$Q \in \mathbb{R}^{n\times n}$	a symmetric matrix
$r_k,\ k = 0,1,\ldots$	penalty parameters (positive) in penalty function methods
$r_k = \nabla f\left(x_{k+1}\right) - \nabla f\left(x_k\right) = g_{k+1} - g_k$	(in the DFP method)
s_b and s_n	step sizes for basic and non-basic variables in Example 2.7
s_k	step size at the k^{th} iteration
s_{min}, s_{max}	minimum and maximum step sizes in Example 2.7
$s_i, i = 1,2,3$	components of the search direction (Example 2.8)
S_Y	strength variable in stress units (Example 2.8)
T	axial load in Example 2.8

$T_j^e, j = 12,3$	nodal temperatures of an element in the FE (finite element) model.
$T \in \mathbb{R}^m$	temperature vector in the FE model (Example 2.1)
$\mho$ and $\delta\mho$	domain and its boundary
U	vector of nodal displacements (Example 2.2)
x	vector of design variables
x_b	basic design variables in the LP problem
x_n	non-basic design variables in the LP problem
x^*	optimum solution
$x_{l,n}$, $x_{u,n}$	lower and upper bounds for the non-basic variables in the LP problem
$y = \{y_j, j = 1,2,\ldots\}$	vector of slack variables in the augmented Lagrangian method and in Example 2.8
z_0	starting vector in terms of standard normal (Example 2.8)
α	parameter to be optimized in the sub-optimization problem (Example 2.8)
β	reliability index (Example 2.8)
β_k	parameter in the CG method (Equations 2.30 and 2.36)
$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$	the Laplacian operator in 2D
$\lambda_i, i = 1,2,\ldots$	Lagrangian multipliers associated with inequality constraints
$\mu_i, i = 1,2,\ldots$	Lagrangian multipliers associated with equality constraints
ρ	mass density (Example 2.2)
σ_A	standard deviation of the RV A (Example 2.8)
σ_{S_Y}	standard deviation of the RV S_Y (Example 2.8)
σ_T	standard deviation of the RV T (Example 2.8)
$\psi(x)$	penalty function (Equation 2.88)
$\nabla f(x)$	gradient of $f(x)$ and direction of steepest ascent
$\nabla_x h$	gradient of $h(x)$, an equality constraint with respect to x
∇_x	gradient of the Lagrangian with respect to x
∇_x^2	Hessian of the Lagrangian with respect to x

EXERCISES

1. For the non-quadratic Rosenbrock function, $f(x_1, x_2) = 100\left(x_1^2 - x_2\right)^2 + \left(x_1 - 1\right)^2$ apply Newton's method. Use the starting point as $(5,5)^{\mathrm{T}}$.

(Hint: The optimum point $\boldsymbol{x}^* = (1,1)^{\mathrm{T}}$ and $f(\boldsymbol{x}^*) = 0$)

2. Use the BFGS method to find the minimum of the quadratic function $f(x_1, x_2) = (x_1 - 3)^2 + (x_2 - 5)^2 + x_1 x_2$ and of the non-quadratic function (Rosenbrock) $-f(x_1, x_2) = 100\left(x_1^2 - x_2\right)^2 + \left(x_1 - 1\right)^2$ (Hint: $\boldsymbol{x}^* = (0.667, 4.667)^T$ for the quadratic function and $\boldsymbol{x}^* = (1,1)^T$ for the non-quadratic function).

3. A production company has four products whose demand x_i, $\mathrm{i} = 1,2,3,4$ is related to their prices $p_i, i = 1,2,3,4$ as:

$$x_1 + 1.5143p_1 = 2671, x_2 + 0.0203p_2 = 135, x_3 + 0.0136p_3 - 0.0015p_4 = 103 \text{ and}$$

$$x_4 - 0.0016p_3 + 0.0027p_4 = 19 \qquad \text{(E2.1)}$$

Maximize the company's revenue $\sum_{\mathrm{i}=1}^{4} x_i p_i$ subjected to the production constraints:

$$0.026x_1 + 0.8x_2 + 0.306x_3 + 0.245x_4 \le 121$$

$$0.086x_1 + 0.02x_2 + 0.297x_3 + 0.371x_4 \le 250$$

$$x_i, p_i, i = 1,2,3,4 > 0 \qquad \text{(E2.2)}$$

(Ravindran *et al*. 2006)

[**Hint**: Using the demand-price relationships, the objective function may be reduced to a quadratic function in single variable x_i or p_i. Solve the resulting optimization problem by any of the derivative methods.]

4. Minimize $f(\boldsymbol{x}) = 0.5\left(x_1^2 + \frac{1}{3}x_2^2\right)$ subjected to the linear constraint $x, x_2 = 1$.

Here $\boldsymbol{x} = \left(x_1, x_2\right)^T$. [Hint: If augmented Lagrangian method is used to solve the problem, Equations (2.117–119) yield (Bertsekas 1996):

$$x_{1,k+1} = \frac{r_{k+1} - \lambda_{k+1}}{1 + 4r_{k+1}},\ x_{2,k+1} = \frac{3\left(r_{k+1} - \lambda_{k+1}\right)}{1 + 4r_{k+1}} \text{ and}$$
$$\lambda_{k+1} = \lambda_k + r_{k+1}\left(x_1 + x_2 - 1\right) \tag{E2.3}$$

The optimum point $\boldsymbol{x}^*$ is $\left(0.25,\ 0.75\right)^{\mathrm{T}}$]

5. Use the augmented Lagrangian along with descent methods (SDM or CG) to minimize $f(\boldsymbol{x}) = x_1 x_2$ subjected to the constraint $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = 1$ where $\boldsymbol{x} = \left(x_1, x_2\right)^{\mathrm{T}}$.

6. Minimize the Rayleigh quotient $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}$ under the constraint $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = 1$ by the augmented Lagrangian method (see also Exercise 8 in Chapter 1).

7. Obtain the solution of the following optimization problem by .Zoutendijk's method of feasible directions:

minimize $f\left(x_1, x_2\right) = -x_1 x_2$

$$\text{s.t. } 8 - x_1 \geq 0, 1 - x_2 \geq 0, x_1 \geq 0, x_2 \geq 0 \tag{E2.4}$$

Show that the solution fails to converge and has an oscillatory behaviour.

8. Solve the following problem by the Zoutendijk's method of feasible directions:

$$\text{minimize } f(\boldsymbol{x}) = 2x_1^2 + 2x_2^2 - 2x_1 x_2 - 4x_1 - 6x_2$$
$$\text{s.t. } x_1 + 5x_2 \leq 5, 2x_1^2 - x_2 \leq 0, -x_1 \leq 0, -x_2 <= 0 \tag{E2.5}$$

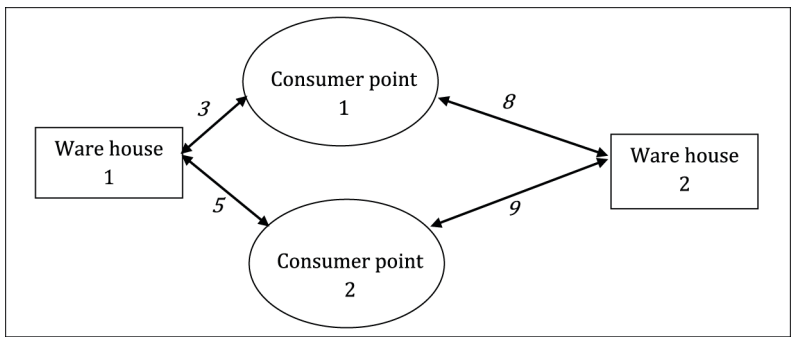


FIGURE E2.1 Optimization problem of minimizing vehicle transportation costs from warehouses to the consumer locations.

Take the starting point as $x_0 = (0,\ 0.75)^T$. Solve the problem by the modified algorithm of Topkis and Veinott also.

9. Solve the optimization problem in Exercise 3 by simplex method of LP programming.

10. Two warehouses need to supply material to two consumer points located as shown in the Figure E2.1. Availability of vehicles at warehouse 1 is 6 and it is 8 at warehouse 2. Given the distances in kilometres from the supply to consumer points, it is required to optimize the travel costs of transportation of material from the warehouses to the consumer points. Number of vehicles required per day to the consumer point 1 is 4 and it is 7 by the second one. Solve the optimization problem by simplex method of LP programming.

[**Hint**: If it is assumed that x_1 and x_2 are the number of vehicles employed by the warehouse 1 and x_3 *and* x_4 by the warehouse 2 to each of the two consumer points, the optimization problem is:

$$\text{minimize: } 3x_1 + 5x_2 + 8x_3 + 9x_4$$

$$\text{s.t. } x_1 + x_2 \le 6, x_3 + x_4 \le 8, x_1 + x_3 = 4, x_2 + x_4 = 7 \text{ and } x_i \ge 0 \qquad \text{(E2.6)}$$

By use of the two equality constraints, the problem may be reduced to a 2-dimensional problem, say, in terms of x_3 *and* x_4 subjected to the constraints:

$$x_3 + x_4 \le 8, -x_3 - x_4 \le -5, x_3 \le 4 \text{ and } x_4 \le 7, x_4 \le 7 \qquad \text{(E2.7)}$$

and the optimal solution is $x_1 = 4,\ x_2 = 2,\ x_3 = 0,\ x_4 = 5$ and the minimum possible distance = 57 km.

A graphical solution is also possible for this two-dimensional LP problem.]

11. Use exterior penalty function method to solve the constrained optimization problem in Equation (2.128) involving the Rosen-Suzuki function.

12. Consider the Camelback function (Molga and Smutnicki 2005):

$$f(x, y) = \left(4 - 2.1x^2 + x^{4/3}\right)x^2 + xy + \left(-4 + 4y^2\right)y^2 \qquad \text{(E2.8)}$$

With $-1.5 \le x \le 1.5$ and $-2 \le y \le 2$, the Camelback function has six minima out of which two are global minima $(-0.0898, 0.7126)$ and $(0.0898, -0.7126)$ with the function value equal to -1.0316. Solve for the minima using SQP.

13. Solve the optimization of the Rosen-Suzuki function (Belegundu and Chandrupatla 1999) by generalized reduced gradient method (Section 2.6 of Chapter 2):

$$\text{min imize } f(\boldsymbol{x}) = x_1^2 + x_2^2 + 2x_3^2 - x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4 + 100 \quad \text{(E2.9)}$$

s. t.

$$g_1(\boldsymbol{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \le 0$$

$$g_2(\boldsymbol{x}) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \le 0$$

$$g_3(\boldsymbol{x}) = 2x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 - 5 \le 0$$

$$\text{and } x_l(i) = -100 \le x_{1,} x_2, x_3, x_4 \le 100 = x_u(i), i = 1,2,3,4. \quad \text{(E2.10)}$$

[**Hint:** Steps involved in the first iteration are enumerated below along with the final solution. Readers may refer to Belegundu and Chandrupatla (1999) for details.

With the slack variables x_5, x_6 and x_7 added to the inequality constraints $g_i(\boldsymbol{x})$, i = 1,2,3, the vector $\boldsymbol{x}$ is $\left(x_{1,} x_2, x_3, x_4, x_5, x_6, x_7\right)^T$. Hence n = 7 and m = 3. The resulting equality constraints are now represented by:

$$h_1(\boldsymbol{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 + x_5 - 8 = 0$$

$$h_2(\boldsymbol{x}) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 + x_6 - 10 = 0$$

$$h_3(\boldsymbol{x}) = 2x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 + x_7 - 5 = 0$$

$$\text{with } x_5, x_6, x_7 \ge 0 \quad \text{(E2.11)}$$

Let us start the iterations with the feasible point $\hat{\boldsymbol{x}} = (0,0,0,0,8,10,5)^T$ where $h_i(\hat{\boldsymbol{x}}) = 0$, i = 1,2,3. With the starting value of $f(\hat{\boldsymbol{x}}) = 100$, the gradient vector $\nabla_x f$ is:

$$\nabla_x f = \frac{\partial f}{\partial \boldsymbol{x}} = \left(2x_1 - 5,\ 2x_2 - 5,\ 4x_3 - 21,\ -2x_4 + 7,\ 0,\ 0,\ 0\right)^T \quad \text{(E2.12)}$$

And

$$\nabla_x \boldsymbol{h} = \begin{bmatrix} 2x_1 + 1 & x_2 - 1 & 2x_3 + 1 & 2x_4 - 1 & 1 & 0 & 0 \\ 2x_1 - 1 & 4x_2 & 2x_3 & 4x_4 - 1 & 0 & 1 & 0 \\ 4x_1 + 2 & 2x_2 - 1 & 2x_3 & -1 & 0 & 0 & 1 \end{bmatrix} \quad \text{(E2.13)}$$

TABLE E2.1
Gradient Vectors of the Objective Function and Constraints at the Feasible Point $\hat{x}$ in the First Iteration

	$B = \frac{\partial h}{\partial x_b} = \nabla_{x_b} h$			$C = \frac{\partial h}{\partial x_n} = \nabla_{xn} h$			
	Basic variables			**Non-basic variables**			
$\nabla_x h$	x_1	x_4	x_3	x_2	x_5	x_6	x_7
$A = [B \quad C]$	$\begin{bmatrix} 1 & -1 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & -1.5 \end{bmatrix}$			$\begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ 0.5 & -1.5 & 0.5 & 1 \end{bmatrix}$			
$\nabla_x f$	$\frac{\partial f}{\partial x_b} = \nabla_{x_b} f$			$\frac{\partial f}{\partial x_n} = \nabla_{x_n} f$			
	$\begin{bmatrix} -5 & 7 & -21 \end{bmatrix}^T$			$\begin{bmatrix} (-5, & 0, & 0, & 0)^T \end{bmatrix}$			

Iterations start with the selection of the basic and non-basic variables by the pivoting operation on $A = \nabla_x$. Table E2.1 shows these variables along with the vector $\nabla_x f$ and the matrix A evaluated at $x = \hat{x}$.

Thus $x_b = (x_1, x_4, x_3)^T$ and $x_n = (x_2, x_5, x_6, x_7)^T$. The reduced gradient from Equation (2.157) is:

$$\frac{d\hat{f}}{dx_n} = \left[-\left(\frac{\partial f}{\partial x_b}\right)^T B^{-1} C + \left(\frac{\partial f}{\partial x_n}\right)^T \right]^T = \begin{pmatrix} 16 & -21 & 4 & 10 \end{pmatrix}^T \quad \text{(E2.14)}$$

The steepest descent direction d_n for x_n is $-\frac{d\hat{f}}{dx_n}$. Equation (2.156b) gives the descent direction d_b for the basic variables as $d_b = -B^{-1} C d_n = \begin{pmatrix} 3.333 & 0.6667 & 34.333 \end{pmatrix}^T$. To proceed with the unconstrained optimization in the original scenario of n variables, we use $d = \begin{pmatrix} d_b^T & d_n^T \end{pmatrix}^T$ for a line search. With the specified lower and upper bounds x_l and x_u for x in Equation (E2.11), it is convenient to fix a suitable step size $s > 0$. That is, with $x_0 = \hat{x}$, the maximum step size s_n for the non-basic variables with respect to $x_{l,n}$ and $x_{u,n}$ may be fixed as:

$$s_n = \frac{x_{l,n}(i) - x_{0,n}(i)}{d_n(i)}, \text{if } d_n(i) < 0 \text{ and}$$

$$s_n = \frac{x_{u,n}(i) - x_{0,n}(i)}{d_n(i)}, \text{if } d_n(i) > 0 \tag{E2.15}$$

Similarly the maximum step size s_b for the basic variables is estimated. The smaller of s_b and s_n may be taken as the maximum step size $s_{\max}$ and the update is obtained with $\boldsymbol{d}_0 = \boldsymbol{d}$ as:

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + s_{max}\boldsymbol{d}_0 \tag{E2.16}$$

The update is accepted for the next iteration provided $f(\boldsymbol{x}_1) < f(\boldsymbol{x}_0)$. In the present problem, $\min(s_b, s_n) = 0.381$. Thus with $s_{max} = 0.381$, $\boldsymbol{x}_1$ as obtained from (E2.16) gives $f(\boldsymbol{x}_1) = 171.125 > f(\boldsymbol{x}_0) = 100$. The update is obviously not acceptable and a line search is performed to find a suitable step size $s \in [0, s_{max}]$ such that an unconstrained minimum is obtained. With $s = 0.154$ obtained by line search – using golden section method (see Exercise 1, Chapter 1) – the new $\boldsymbol{x}_1$ is:

$$\boldsymbol{x}_1 = (0.516, 2.478, 5.318, 0.103, 4.747, 10.61, 6.54)^T \tag{E2.17}$$

The update gives $f(\boldsymbol{x}_1) = 37.03 < f(\boldsymbol{x}_0)$ and is acceptable. Yet, the update needs a check for feasibility before proceeding to the next iteration. The feasibility requirement is to see that $\max_i \{h_i(\boldsymbol{x}_1)\}, i = 1,2,3$ is less than, say, $\mu = 10^{-4}$. With $max_i \{h_i(\boldsymbol{x}_1)\} = 40.8$ in this case, $\boldsymbol{x}_1$ is not feasible. Here, a strategy is adopted to make it feasible by keeping $\boldsymbol{x}_n$ fixed whilst suitably varying $\boldsymbol{x}_b = (x_1, x_4, x_3)^T$ so that the feasibility condition $h_i(\boldsymbol{x}_b, \boldsymbol{x}_n), i = 1,2,3 \cong \mu$ is satisfied (see Figure 2.24). This is equivalent to solving the nonlinear equations $h(\boldsymbol{x}_b \boldsymbol{x}_n) = 0$ for $\boldsymbol{x}_b$ by a Newton-Raphson method. Starting with $\boldsymbol{x}_{b,0} = (0.516,\ 0.103,\ 5.318)^T$, the method iteratively yields $\boldsymbol{x}_{b,new}$ as:

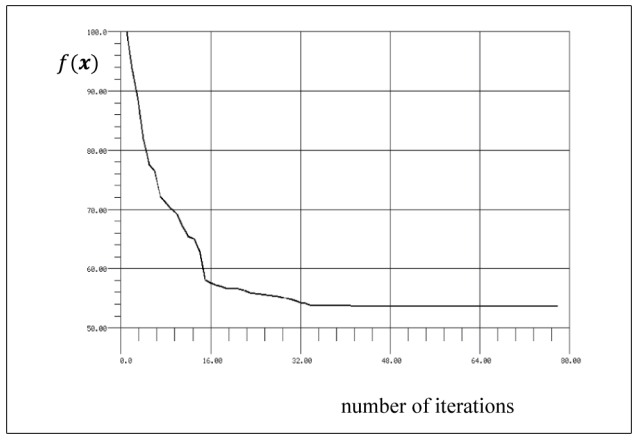


FIGURE E2.2 Minimization of Rosen-Suzuki function by GRG method, evolution of objective function with iterations with final minimum value = 53.64.

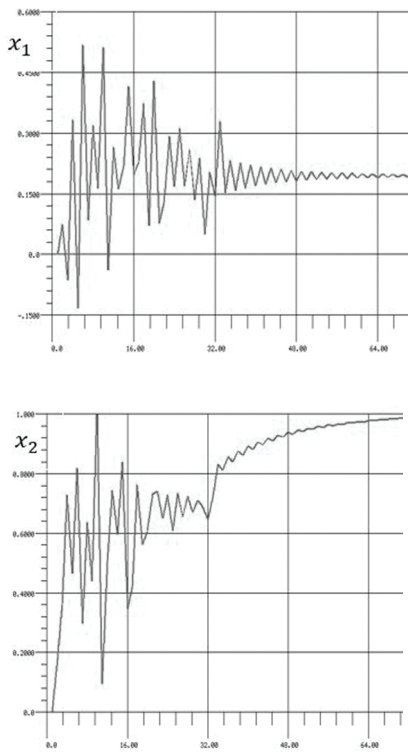
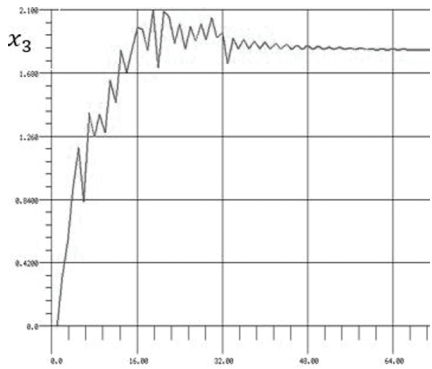


FIGURE E2.3a–b Minimization of Rosen-Suzuki function by GRG method, evolution of design variables x_1, x_2, x_3 and x_4 with iterations.



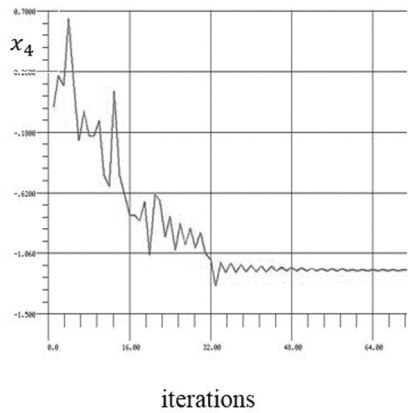


FIGURE E2.3c–d (Continued)

$$\boldsymbol{x}_{b,new} = \boldsymbol{x}_{b,new} + \Delta \boldsymbol{x}_b \tag{E2.18}$$

where

$$\Delta \boldsymbol{x}_b = -J^{-1}\boldsymbol{h}\left(\boldsymbol{x}_{b,}\boldsymbol{x}_n\right) \tag{E2.19}$$

$J \in \mathbb{R}^{m\times m}$ is the Jacobian matrix given by the partial gradient of $\boldsymbol{h}$ with respect to $\boldsymbol{x}_b$. The Newton-Raphson method converges with $max\ h_i\left(\boldsymbol{x}_1\right) < \mu, i = 1,2,3$ in five iterations. In fact, the method has failed to converge in the initial stages and the step size s_{max} is progressively reduced till convergence is realized. With the above correction to $\boldsymbol{x}_b$ and the feasibility condition satisfied; it marks the end of a successful iteration. Figure E2.2 shows the evolution of the objective function with iterations and its final minimum value $f\left(\boldsymbol{x}^*\right) = 53.64$. The optimum point to the Rosen-Suzuki

function is $\left(x_1^*, x_2^*, x_3^*, x_4^*\right) = \left(0.197, 0.99, 1.83, -1.19\right)$ whose convergence with iterations is shown in Figure E2.3.

REFERENCES

Abadie, J. and J. Carpentier. 1969. Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints. In: *Optimization*. R. Fletcher (ed.), pp. 37–47. Academic Press.

Ang, A. H. S. and W. H. Tang. 1975. *Probability Concepts in Engineering Planning and Design, Basic Principles.* John Wiley & Sons. Inc. NY.

Ang, A. H. S. and W. H. Tang. 1984. *Probability Concepts in Engineering Planning and Design, Vol. 2:* Decision, Risk, and Reliability. 1st Ed. John Wiley & Sons. Inc. NY.

Augarde C. E., A. Ramage, and J. Staudacher. 2006. An element-based displacement preconditioner for linear elasticity problems. *Computers & Structures* 84(31–32): 2306–2315.

Bartlett, M. S. 1951. An inverse matrix adjustment arising in discriminant analysis. *Annals of Mathematical Statistics* 22(1): 107–111.

Bathe, K. J. 1996. *Finite Element Procedures*. Prentice-Hall International, Inc., Englewood Cliffs, NJ, USA.

Bazaraa, M. S., J. J. Jarvis, and H. D. Sherali. 1990. *Linear Programming and Network Flows.* 2nd Ed., Wiley. NY.

Bazaraa, M. S., H. D. Sherali, and C. M. Shetty. 2006. *Nonlinear Programming*. John Wiley & Sons. Inc., NJ.

Belegundu, A. D. and T. R. Chadrupatla. 1999. *Optimization Concepts and Applications in Engineering*. Prentice Hall.

Benzi, M, 2002. Preconditioning techniques for large linear systems: a survey. *Journal of Computational Physics* 182: 418–477.

Bertsekas, D. P. 1996. *Constrained Optimization and Lagrange multiplier Methods*. Athena Scientific, Belmont, USA. (Originally published by Academic Press. Inc. in 1982)

Boyd, S. and L. Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press, NY.

Broyden, C. G. 1967. Quasi-Newton methods and their application to function minimization. *Mathematics of Computation* 21: 368–381.

Broyden, C. G. 1970. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications* 6: 76–90.

Cauchy. A. 1847. Methodegenerale pour la resolution des systemes d'equations simultanes. *Comptes rendus de l'Académie des Sciences* Paris 25: 536–538.

Chan, T. F. and H. A. van der Vorst. 1997. Approximate and incomplete factorizations. In: *Parallel Numerical Algorithms*, 4, 167, edited by D. E. Keyes, A. Sameh, and V. Venkatakrishnan, ICASE/LaRC Interdisciplinary Series in Science and Engineering, Kluwer Academic. Dordrecht.

Concus, P. and Golub, G. H. 1976. *A Generalized Conjugate Gradient Method for Nonsymmetric Systems of Linear Equations*. Computer Science Department, Stanford University.

Conn, A., N. I. M. Gould and P. L. Toint. 1991. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis* 28: 545–572.

Cook, R. D., D. S. Malkus and M. E. Plesha. 1989. *Concepts and Applications of Finite Element Analysis*. John Wiley, NY

Curry, H. D.1944. The method of steepest descent for non-linear minimization problems. *Quarterly of Applied Mathematics* 2: 258.

Dantzig, G. B. 1951. *Application of the Simplex Method to a Transportation Problem, Activity Analysis of Production and Allocation, Cowles Commission Monograph No. 13*. T. C. Koopmans, Ltd., John Wiley and Sons, Inc., NY.

Dantzig, G. B. and J. H. Ramser. 1951. *The Truck Dispatching Problem, Management Science.* Lenstra: Rinnooy Kan & Schrijver

Dantzig, G. B., 1963, *Linear Programming and Extensions*. Princeton Univ. Press, Princeton, NJ.

Dantzig, G. B. and M. N. Thapa. 2003. *Linear Programming. 2: Theory and Extensions*. Springer. USA.

Datta. B. N. 2010. *Numerical Linear Algebra and Applications*. SIAM, Philadelphia, 2 Ed.

Davidon, W.C. 1959. *Variable Metric Method for Minimization*, R&D Report ANL – 5990, U.S. Atomic Energy Commission, Argonne National Laboratories. (Reprinted in *SIAM Journal of Optimization* 1991; 1: 1–17.)

Dowling, E. 1991. *Introduction to Mathematical Economics*. McGraw-Hill, NY.

Fiacco, A. V. and McCormick, G. P. 1964. The sequential unconstrained minimization technique for nonlinear programming: a primal method. *Management Science* 10: 360–364.

Fiacco, A. V. and G. P. McCormick. 1968. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley and Sons, NY. (Republished by SIAM. Philadelphia; 1990.)

Fletcher, R. 1970. A new approach to variable metric algorithms. *Computer Journal* 13(3): 317–322.

Fletcher, R. 1976. Conjugate gradient methods for indefinite systems. In: *Proceedings of Dundee Conference on Numerical Analysis—1975* (Ed. G. A. Watson). Springer Verlag. Berlin.

Fletcher, R. and M. J. D. Powell. 1963. A rapidly convergent descent method for minimization. *Computer Journal* 6: 16–168.

Fletcher, R. and C. M. Reeves. 1964. Function minimization by conjugate gradients. *Computer Journal* 7: 149–154.

Florian, M. S., S. Nguyen, and S. Pallottino. 1981. A dual simplex algorithm for finding all shortest paths. *Networks* 11: 367–378.

Floudas, C. A. and P.M. Pardalos. 1990. *A Collection of Test Problems for Constrained Global Optimization Algorithms*. Springer-Verlag. Berlin.

Fox, T. 1982. *Nonlinear Optimization With Linear Constraints Using a Projection Method*. NASA Technical Paper 2086.

Gärtner, B. and J. Matousek. 2006. *Understanding and Using Linear Programming*. Berlin: Springer.

Golub, G. H. and C. F. Van Loan. 1996. *Matrix Computations*. JHU Press.

Goldfarb, D. 1970. A family of variable metric updates derived by variational means. *Mathematics of Computation* 24(109): 23–26.

Goldfarb, D. 1985. Efficient dual simplex algorithms for the assignment problem. *Mathematical Programming* 33: 187–203.

Haug, E. J., and J. S. Arora. 1979. *Applied Optimal Design: Mechanical and Structural Systems*. John Wiley. NY.

Hestenes, M. R. 1969. Multiplier and gradient methods. *Journal of Optimization Theory and Applications* 4: 303–320.

Hestenes, M. R. and E. Stiefel. 1952. Methods of conjugate gradients for solving linear systems. Journal of Research of the N*ational Bureau of Standards* 49: 409.

Hildebrand, F. B. 1968. *Finite-Difference Equations and Simulations. Section 2.2.* Prentice-Hall. Englewood Cliffs. NJ.

Hillier, F. S. and G. J. Lieberman. 1990. *Introduction to Mathematical Programming*. McGraw Hill, NY.

Hillier, F. S. and G. J. Lieberman. 1995. *Introduction to Operation Research*. McGraw-Hill. NY.

Hitchcock, F. L. 1941. The distribution of a product from several sources to numerous localities. *Journal of Mathematical Physics* 20: 224–230.

Hughes, T. J. R. and Ferencz. R. M.,1988. Fully vectorized EBE preconditioners for nonlinear solid mechanics: Applications to large-scale three-dimensional continuum, shell and contact/impact problems. In: R. Glowinski et al. (eds.) *Domain Decomposition Methods for Partial Differential Equations*, pp. 261–280. SIAM. Philadelphia. PA.

Hughes, T. J. R., Levit, I. and Winget, J. 1983. An element-by-element solution algorithm for problems of structural and solid mechanics. *Computer Methods in Applied Mechanics and Engineering* 36: 241–254.

Iusem, A. N. 2003. On the convergence properties of the projected gradient method for convex optimization. *Computational and Applied Mathematics* 22(1): 37–52.

Jennifer B. E. and F. M. Roummel. 2017. On solving large-scale limited-memory quasi-Newton equations. *Linear Algebra and its Applications* 515: 196–225.

Karmarkar, N., 1984. A new polynomial-time algorithm for linear programming. *Combinatorica* 4(4): 373–395.

Kantorovich, L. 1942. On the translocation of masses., *DokladyAkad. Nauk SSSR* 37: 199–201.

Kershaw, D. S. 1978. The incomplete Cholesky conjugate gradient method for the iterative solution of systems of linear equations. *Journal of Computational Physics* 26: 43.

Kreyszig, E. 1999. *Advanced Engineering Mathematics*. John Wiley & Sons. Inc.

Lasdon, L. S., R. L. Fox, and M. W. Ratner. 1973. *Nonlinear Optimization Using the Generalized Reduced Gradient Method. Technical Memorandum No. 325*. Office of Naval Research. Springfield, USA.

Lasdon, L. S., A. D. Waren, A. Jain, M. Ratner. 1978. Design and testing of a generalized reduced gradient code for nonlinear programming. *ACM Transactions on Mathematical Software* 4(1): 35–50.

Lemke, C. E. 1954. The dual method of solving the linear programming problem. *Naval Research and Logistics Quarterly* 1: 36–47.

Liu, G. R. and S. S. Quek. 2003. *The Finite Element Method: A Practical Course*. NY: Elsevier Science Ltd.

Makinson, G. H. and Shah, A. A., 1986. An iterative solution method for solving sparse nonsymmetric linear systems. *Journal of Computational and Applied Mathematics* 15: 339–352.

Maymon, G. 1998. *Some Engineering Applications in Random Vibrations and Random Structures*. AIAA Inc. Virginia. USA.

Meijerink, J. A. and H. A. van der Vorst. 1977. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of Computation* 31: 148.

Melchers, R. E. 2007. *Structural Reliability Analysis and Prediction*. 2nd Ed. John Wiley & Sons.

Molga, M. and C. Smutnicki. 2005. *Test Functions for Optimization Needs*. Retrieved June 2013, from www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf.

Murty, K. G. 1983. Linear Programming. John Wiley & Sons. Inc. NY.

Nocedal, J. and S. J. Wright. 2006. Numerical Optimization. 2nd Ed. Springer-Verlag, NY.

Ogata, K. 1995. *Discrete-time Control Systems*. Prentice-Hall. Inc. 2nd Ed. NJ.

Oliveira, A. and Sorensen. D. 1997. *A New Class of Preconditioners for Large-scale Linear Systems from Interior Point Methods for Linear Programming*. Technical Report CRPC-TR97771, Center for Research on Parallel Computation, Rice University.

Powell, M. J. D. 1969. A method for nonlinear constraints in minimization problems. In: *Optimization*, R. Fletcher (ed.), pp. 283–298. Academic Press. NY.

Powell, M.J.D. 1978. A fast algorithm for nonlinearly constrained optimization calculations. *Lecture Notes in Mathematics*. no. 630. Springer-Verlag, NY.

Quarteroni, A. and Valli, A. 1999. *Domain Decomposition Methods for Partial Differential Equations*. Clarendon. Oxford.

Rao, S. S. 1984. *Optimization: Theory and Applications*. 2nd Ed. John Wiley & Sons. NY.

Ravindran, A., K. Ragsdell, and G. Reklaitis. 2007. *Engineering Optimization: Methods and Applications*. New Delhi: Wiley-India.

Rockafellar, R.T. 1974. Augmented Lagrange multiplier functions and duality in non-convex programming. *SIAM Journal on Control and Optimization* 12(2): 268–285.

Roos, C., T. Terlaky, and J. Vial. 2006. *Interior Point Methods for Linear Optimization*, 2nd Ed. Springer-Verlag.

Rosen, J. B. 1960. The gradient projection method for nonlinear programming. Part I. Linear constraints. *SIAM Journal on Applied Mathematics* 8: 181–217.

Rosen, J. B. 1961. The gradient projection method for nonlinear programming, Part 2: Nonlinear constraints. *SIAM Journal on Applied Mathematics* 9: 514–553.

Schlttkowski, K. 1982. *On The Convergence Of A Sequential Quadratic Programming Method With An Augmented Lagrangian Line Search Functions*. TR. SOL 82-4. Dept. of operations research. Stanford University. CA.

Shanno, D. F. 1970. Conditioning of quasi-Newton methods for function minimization *Mathematics of Computation* 24(111): 647–656.

Smith, B. F., Bjørstad, P. E. and Gropp, W. D., 1996. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge Univ. Press. UK.

Stigler, G. 1945. The cost of subsistence. *Journal of. Farm Economics* 25: 303–314.

Tezduyar, T. E. and Liou, J. 1989. Grouped element-by-element iteration schemes for incompressible flow computations. *Computer Physics Communications* 53: 441–453.

Topkis, D. M. and Veinott, A. F. 1967. On the convergence of some feasible direction algorithms for nonlinear programming. *SIAM Journal on Control and Optimization* 5: 268–279.

Vanderplaats, G.N. 1973. *CONMIN – A FORTRAN Program for Constrained Function Minimization User's Manual*, NASA Ames Research Center Technical Memorandum. NASA TM X-62. 282.

Vanderplaats, G.N. 1984. An efficient feasible directions algorithm for design synthesis. *AIAA Journal* 22(11): 633–640.

Watkins. D. S. 2010. *Fundamentals of Matrix Computations*, 3rd edn. Wiley.

Wolfe, P. 1959. The simplex method for quadratic programming. *Econometrica* 27: 382–398.

Wolfe, P. 1963. Methods of nonlinear programming. In: R.L. Graves, P. Wolfe (Eds.), *Recent Advances in Mathematical Programming*. McGraw-Hill. NY.

Yan, L. and D. Ma. 2001. Global optimization of non-convex nonlinear programs using line-up competition algorithm. Computers & Oper*ations Research* 25(11–12): 1601–1610.

Zoutendijk, G. 1960. *Methods of Feasible Directions*. Elsevier. Amsterdam, 1960.

BIBLIOGRAPHY

Hillier, F. S. and Gerald J. Leberman. 1986. *Introduction to Operations Research*, 4th ed., Oakland, Calif.: HoldenDay.

Hillier, F. S. and G. J. Lieberman. 1995. *Introduction to Operation Research*. NY: McGraw-Hill.

Hughes, T. J. R. 1987. *The Finite Element Method*. Prentice-Hall International, Inc. Englewood Cliffs. NJ. USA.

Lanczos, C. 1951. *Solution of systems of linear equations by minimized iterations*, NAML Report 52-13. National Bureau of Standards.

Leissa, A. W. 2005. The historical bases of the Rayleigh and Ritz methods. *Journal of Sound and Vibration* 287: 961–978.

Morgan, S. S. 1997. *A Comparison of Simplex Method Algorithms*, Master's thesis, University of Florida.

Noor, A. K. 1991. Bibliography of books and monographs on finite element technology. *Applied Mechanics Reviews* 44(6): 307–317.

Ravindran, A., Don. T. Philips, and James J. Solberg. *Operations Research: Principles and Practice*, 2nd ed., New York: Wiley, 1986.

Saad Y. and M.H. Schultz. 1986. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific Computing* 7: 856–869.

Winston, W. L. 1991. *Operations Research: Applications and Algorithms*, 2nd ed., Boston: PWS-Kent.

Yinyu Ye, 1997. *Interior Point Algorithms: Theory and Analysis*, (advanced graduate-level). Wiley.

3 Classical Derivative-free Methods of Optimization

3.1 INTRODUCTION

In this chapter, we discuss the broad class of derivative-free methods of optimization (Rios and Sahinidis 2013). They include direct search methods (Hooke and Jeeves 1961, Nelder and Mead 1965, Kelley 1999) and also evolutionary methods such as the genetic algorithm (Sotiropoulos et al. 1997), ant colony optimization (Dorigo et al. 2011), simulated annealing (Van Laarhoven and Aarts 1987), particle swarm optimization (Kennedy and Eberhart 1995, Slowik and Kwasnicka 2018, Zhao et al. 2019), differential evolution (Storn and Price 1997). As the name indicates, derivative-free methods employ only function evaluations in the optimization process without the need for derivative computation, unlike derivative-based methods described in the last chapter.

A review article by Lewis et al. (2000) extensively covers direct search methods. The term 'direct search method', first coined by Hooke and Jeeves (1961), explores the feasible space at any iteration for the best solution out of trial solutions and decides the strategy for the next set of trial solutions. They are particularly suited for solving non-convex problems (Walters et al. 1991) that may have many local optima. In the absence of generic directional information, derivative-free methods appear to be a good option to find the best local extremum. Further, the methods are formulated using simple arguments whilst affording flexibility in the computer implementation. The erstwhile perception that direct search methods are mostly heuristic without proofs of convergence does not hold, as shown by many researchers (Conn et al. 1996, Kelley 1999, Lucidi and Sciandrone 2002, Torczon 1991, 1997). With the availability of convergence proofs, there has of late been a significant upsurge in both their encapsulation within efficient computer software and usage. In the literature, the words 'derivative-free', 'direct search' and 'pattern search' are often used interchangeably.

The ineffectiveness or inapplicability of a Newton-type search has also led to the emergence of evolutionary methods, most of which employ stochastic (i.e. random evolutionary) search rooted in a meta-heuristic origin (Holland 1975, Glover and Kochenberger 2003). The prime reason for the popularity of these methods over other optimization techniques is their ability to solve computationally complex decision problems within a reasonable computational time. The stochastic nature of search which is the hallmark of these methods aids in the computed solution getting unlocked from local minima, thus potentially realizing better global solutions. The underlying justification in these methods is often sociological or biological (e.g. Darwinian evolution). Applications of stochastic search methods (Fletcher 1987, Chong and Zak

 DOI: 10.1201/9781003414063-3

2013) include problems with sufficiently smooth, yet multimodal, objective/cost functionals, wherein the use of directional derivatives may be inadequate in obtaining the global optimum. In this context, the derivative-free methods of Hooke and Jeeves (1961) Nelder and Mead (1965) may be included in the deterministic search category. While a stochastic route often facilitates an effective search, the posing of the original problem could itself be deterministic, the aim merely being to arrive at the design point that is the global extremum of an objective functional, possibly subject to a set of prescribed constraints. The present chapter is meant to be a brief repository of this general class of derivative-free methods.

3.2 DIRECT SEARCH METHODS

Apart from the advantage of no computation of derivatives and thus being more effective in dealing with non-smooth functions, these derivative-free methods are also suited to cases where the objective function and constraints are implicitly stated (i.e., not computable through explicit expressions). In these cases, to obtain reliable estimates for derivatives even by finite difference approximations is a tough ask, which may sometimes be infeasible thus rendering derivative-based methods inapplicable. For instance, a requirement of this kind is encountered in finite element-based system design towards achieving, say, maximum reliability or / and cost minimization. Applications may be found in Marsden et al. (2007) for aircraft design, Bartholomew-Biggs et al. (2003) for aircraft routing, Bendsøe and Sigmund (2003 and Guirguis and Aly (2016) for topology optimization, Duvigneau and Visonneau (2004) for hydrodynamic design and Marsden et al. (2008) for medicine.

3.2.1 Method of Hooke and Jeeves (HJ)

The method of Hooke and Jeeves (HJ) is a reliable and perhaps the simplest derivative-free method. With a starting candidate $\boldsymbol{x}_0 = \left\{x_{0,1}, x_{0,2}, \ldots, x_{0,n}\right\} \in \mathbb{R}^n$ for the design variables and a chosen step length $s > 0$, it attempts to reach the optimum via a series of exploratory and pattern search moves. Hence the HJ method is also known as a pattern search method. No derivatives of the objective function $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ are invoked in the optimization process. An exploratory move, as the name indicates, explores for a direction in $\mathbb{R}^n$ which decreases $f(\boldsymbol{x})$. The method conveniently chooses the coordinate basis directions to make the exploratory moves. Suppose that $\boldsymbol{x}_k = \left\{x_{k,1}, x_{k,2}, \ldots, x_{k,n}\right\}$ is the solution at the end of the k^{th} iteration. A new iteration starts with an exploratory move in which i^{th} design variable $x_{k,i}$ for each i is sequentially increased or decreased by a step length s_i in the direction of the i^{th} coordinate axis. Note that the choice of step length for each variable is dependent on the specific problem on hand and may be fixed before the start of iterations. Suppose that, as shown in Figure 3.1a of a two-dimensional case, $\boldsymbol{x}_k$ is moved towards the positive X_1-axis by a step length s_1, i.e., with the variable $x_{k,1}$ increased to $x_{k,1} + s_1$ and the change fails to decrease the objective function. The letter 'F' over the dotted line indicates a failure of the move in the positive direction of the X_1-axis. Then x_k is

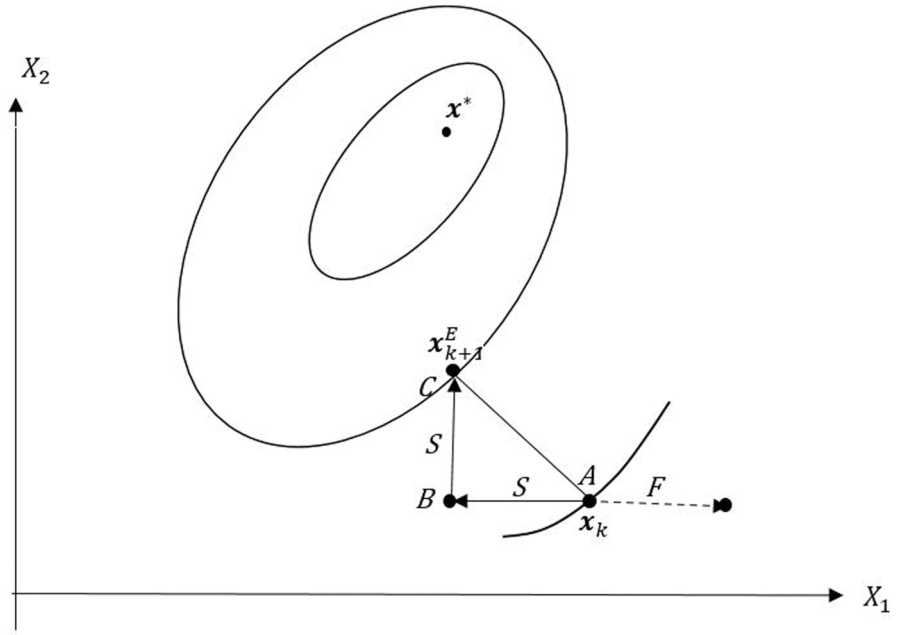


FIGURE 3.1a HJ method – two-dimensional case, an exploratory move, successful step is shown by a dark arrow with the letter '*S*' over or by the side of the line and an unsuccessful step by a dotted arrow with the letter '*F*' over it or by its side.

moved towards the negative X_1-axis by the same step length s_1, i.e. with $x_{k,1}$ decreased to $x_{k,1} - s_1$. Assuming that the move is successful with a decrease in the objective function (as indicated in Figure 3.1a), the change is accepted with $\boldsymbol{x}_k$ moved to the new location B. The procedure is repeated with the variable $x_{k,2}$. In a general n-dimensional case, after all the n directions are exhausted, the exploratory move results in two possibilities. One is that the move is a success with a new point $\boldsymbol{x}^E_{k+1}$ (point C in Figure 3.1a) such that $f\left(\boldsymbol{x}^E_{k+1}\right) < f\left(\boldsymbol{x}_k\right)$. The superscript E stands for the exploratory search. The second possibility is a failure of the exploratory move. In this case, the step size is reduced and the exploratory move repeated using $\boldsymbol{x}_k$ (at the point A). After a successful exploratory move, a pattern move is initiated at $\boldsymbol{x}^E_{k+1}$ (see Figure 3.1b). The pattern move is an 'aggressive' one - an optimistic move with a larger step size in the seemingly successful direction given by $\boldsymbol{x}^E_{k+1} - \boldsymbol{x}_k$. Thus, we move in this direction to obtain $\boldsymbol{x}^P_{k+1} = \boldsymbol{x}_k + c\left(\boldsymbol{x}^E_{k+1} - \boldsymbol{x}_k\right)$ where $1 < c \in \mathbb{R}$. The superscript P stands for the pattern search. Thus, we arrive at D in Figure 3.1b. An exploratory search is now performed at $\boldsymbol{x}^P_{k+1}$ (i.e. the point D). It is shown to be successful in Figure 3.1b with a move in the positive direction of the X_2-axis (i.e., the function value $f\left(\boldsymbol{x}^E\right)$ at the new point E is less than $f\left(\boldsymbol{x}_k\right)$. The pattern search is termed as successful and the k^{th} iteration is complete. A new iteration starts with $\boldsymbol{x}_{k+1} = \boldsymbol{x}^E$. Otherwise (i.e., if

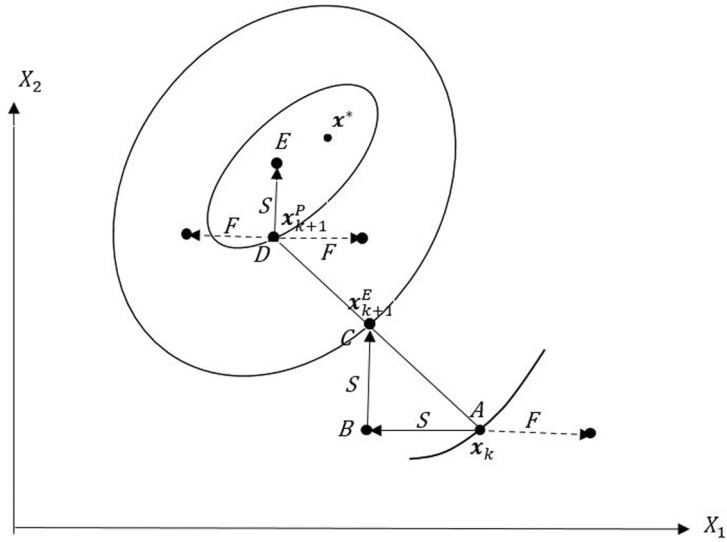


FIGURE 3.1b HJ method – two-dimensional case, a pattern move towards the point D along the direction $x^E_{k+1} - x_k$, the subsequent exploratory move succeeds and reaches the point E and the pattern search is termed as successful.

$f(\boldsymbol{x}^E) \geq f(\boldsymbol{x}_k)$, the pattern search is unsuccessful and the next iteration starts at the point C with $\boldsymbol{x}_{k+1} = \boldsymbol{x}^E_{k+1}$.

The iterations stop when there is no further improvement, decided either by a check on the objective function or on the step size falling below a tolerance level.

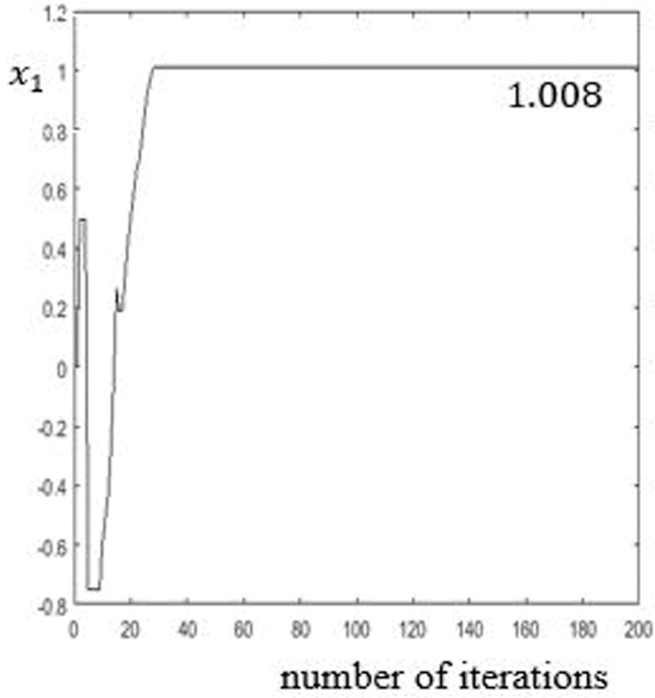
For the Rosenbrock function $f(\boldsymbol{x}) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2$ with $\boldsymbol{x} = (x_1, x_2)^T$, HJ method produces a result as shown in Figures 3.2a–c. In obtaining the result, the lower and upper bounds for both x_1 and x_2 are taken as –10.0 and 10.0, respectively.

Lewis and Torczon (1999, 2000) provided direct search methods with simple bounds (called box-type bounds) on the design variables. We also refer to Lewis and Torczon (2002) for a hybrid approach to handle constrained optimization problems. The following example demonstrates the use of a hybrid method wherein HJ is used along with the interior penalty method (Section 2.4.2, Chapter 2).

Example 3.1. We reconsider the weight optimization of the plane truss Example 2.2 in Chapter 2 by HJ method.

Solution. Refer to Figure 2.12 of Chapter 2 for the FE model of the 10-member plane truss. The constrained optimization problem is defined in Equation (2.109). By the interior penalty method, we form the unconstrained optimization problem which is restated:

$$\text{minmize } \hat{f}(r, \boldsymbol{x}) = f(\boldsymbol{x}) + r\psi(\boldsymbol{x}) \tag{3.1}$$



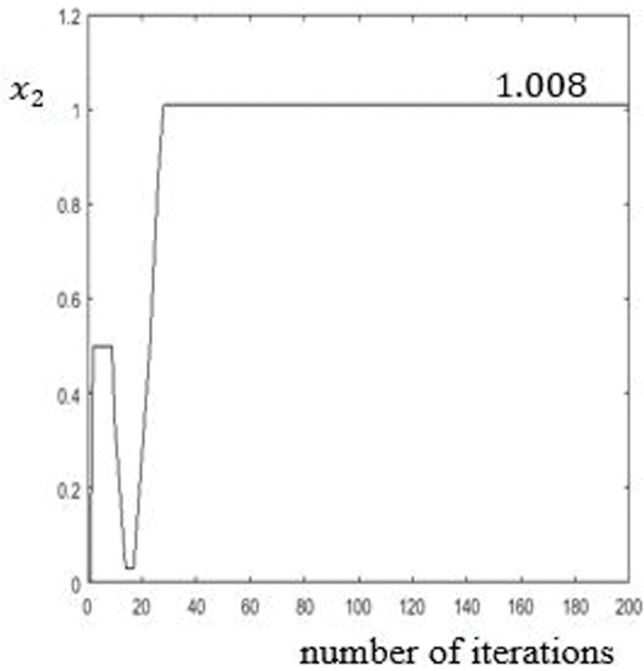


FIGURE 3.2a–b Result for Rosenbrock function (see Figure 2.4, Chapter 2) by HJ method: (a) evolution of x_1, (b) evolution of x_2 with iterations (optimum $x^* = (1.008, 1.008)^T$).

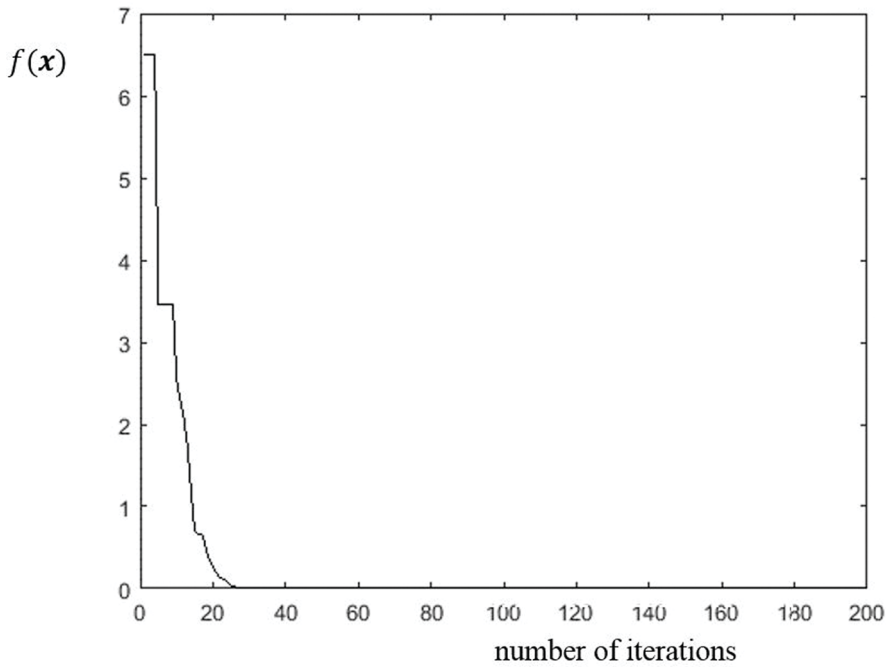


FIGURE 3.2c Result for Rosenbrock function by the HJ method; evolution of the objective function with iterations (finally attaining a minimum value of 0.00626).

where $\boldsymbol{x} = \left(A_i, i = 1, 2, \ldots, 10\right)$ is the vector of cross-sectional areas. The penalty parameter $r > 0$ is associated with the specified constraints with:

$$\psi(\boldsymbol{x}) = -\sum_{i=1}^{10} \frac{1}{A_{l,i} - x_i} - \sum_{i=1}^{10} \frac{1}{x_i - A_{u,i}} - \frac{1}{U_4 - U_b} \tag{3.2}$$

$A_{l,i} = 6$ sq. cm and $A_{u,i} = 10\ sq.\ cm$ are the specified lower and upper bounds for the areas of cross-section. U_4 is the displacement at node 3 in the *Y*-direction. $U_b = 6$ cm is the upper bound on this displacement. Further, as iterations progress, we have a decreasing sequence, $r_k = cr_{k-1}$ with $c \in (0,1)$. At each iteration, the unconstrained optimization problem is solved by the HJ method. (In Example 2.2 of Chapter 2, the CG method is used for the purpose). The result by HJ is shown in Figure 3.3.

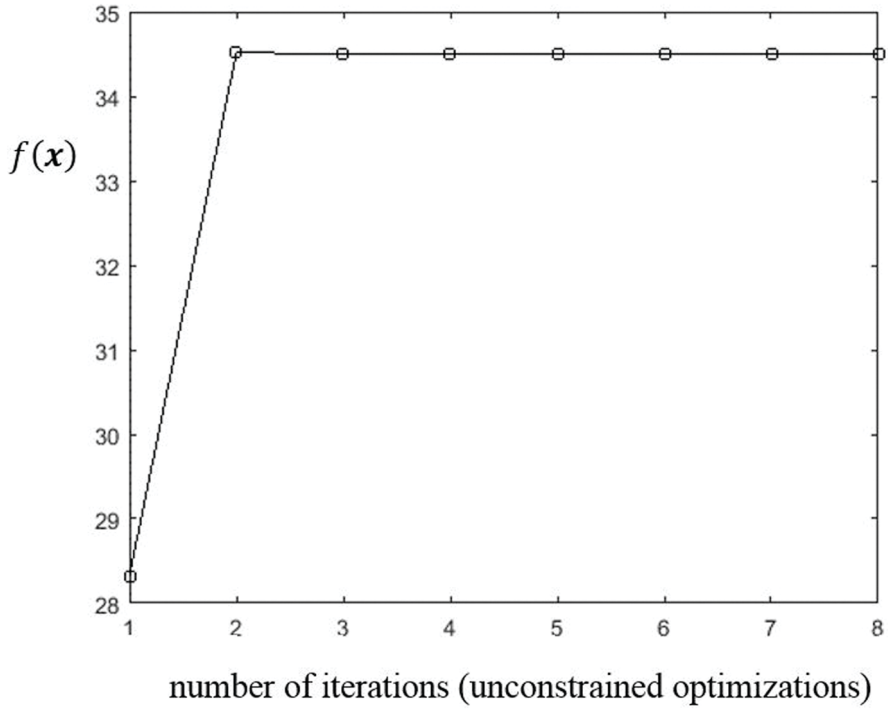


FIGURE 3.3 Weight optimization of a plane truss by HJ combined with the interior penalty function method; $r_0 = 1000$ and $r_k = 0.1 r_{k-1}$, $x_0 = (6,6,6,6,6,6,6,6)^T$, $x^* = (7.3, 6.77, 7.3, 7.7, 6.8, 7.6, 7.3, 7.3, 7.3, 7.6)^T$, Y-displacement at node 3 is $-6.0\ cm$ and optimum weight = 34.5 N at the end of iterations.

■

3.2.2 Simplex Method of Nelder and Mead [NM]

The simplex method of Nelder and Mead (NM) is another direct search method popularly used (Walters et al. 1991, Wright 1996) for multidimensional unconstrained minimization. This simplex method distinctly differs from the simplex method of LP (Chapter 2). The simplex in the present context is a geometric figure which is a convex polytope of n+1 vertices and changes its shape during the iteration process. If $n = 2$, the polytope is a triangle and for $n = 3$, it is a tetrahedron. A tetrahedron is familiarly known as a 3-simplex. Each iteration in the NM method starts with a simplex within the feasible space. The vertices of the simplex constitute the trial solutions. At the k^{th} iteration, let the set of function values evaluated at these trial solutions be ranked in ascending order and the corresponding vertices be $\boldsymbol{x}_{k,1}, \boldsymbol{x}_{k,2}, \ldots, \boldsymbol{x}_{k,n+1}$. Thus $f\left(\boldsymbol{x}_{k,1}\right)$ is the best solution and $f\left(\boldsymbol{x}_{k,n+1}\right)$ the worst.

The method involves movement of the simplex possibly towards a local optimum by virtue of simple geometrical operations (Figures 3.4a–e). The first one is 'reflection'

(Figure 3.4a) which reflects the worst vertex $\boldsymbol{x}_{k,n+1}$ over the centroid $\bar{\boldsymbol{x}}_k$ to the new point $\boldsymbol{x}^R_{k,n+1}$. $\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_{k,i}$ is the centroid of the first n best points of the simplex. In the two-dimensional case, $\bar{\boldsymbol{x}}_k$ is the mid point of the line joining $\boldsymbol{x}_{k,1}$ and $\boldsymbol{x}_{k,2}$.

Reflection (Figure 3.4a)
The reflected point is:

$$\boldsymbol{x}^R_{k,n+1} = \bar{\boldsymbol{x}}_k + \alpha\left(\bar{\boldsymbol{x}}_k - \boldsymbol{x}_{k,n+1}\right),\ \alpha \in \mathbb{R}^+ \tag{3.3}$$

If $f\left(\boldsymbol{x}_{k,1}\right) \le f\left(\boldsymbol{x}^R_{k,n+1}\right) < f\left(\boldsymbol{x}_{k,n}\right)$, accept $\boldsymbol{x}^R_{k,n+1}$ in place of $\boldsymbol{x}_{k,n+1}$ and the iteration is complete.

Two other possibilities may exist: $f\left(\boldsymbol{x}^R_{k,n+1}\right) < f\left(\boldsymbol{x}_{k,1}\right)$ or $f\left(\boldsymbol{x}^R_{k,n+1}\right) \ge f\left(\boldsymbol{x}_{k,n}\right)$.

Expansion (Figure 3.4b)

a) In case $f\left(\boldsymbol{x}^R_{k,n+1}\right) < f\left(\boldsymbol{x}_{k,1}\right)$, it is possible to take an aggressive move via expansion (Figure 3.4b), i.e. move $\boldsymbol{x}^R_{k,n+1}$ with a larger step to get $\boldsymbol{x}^E_{k,n+1}$:

$$\boldsymbol{x}^E_{k,n+1} = \bar{\boldsymbol{x}}_k + \beta\left(\boldsymbol{x}^R_{k,n+1} - \bar{\boldsymbol{x}}_k\right),\ \beta \in \mathbb{R}^+ \tag{3.4}$$

If the resulting $f\left(\boldsymbol{x}^E_{k,n+1}\right) < f\left(\boldsymbol{x}^R_{k,n+1}\right)$, accept $\boldsymbol{x}^E_{k,n+1}$ in place of $\boldsymbol{x}_{k,n+1}$. The iteration is complete.

If $f\left(x^E_{k,n+1}\right) \ge f\left(x^R_{k,n+1}\right) x \ge$, accept $\boldsymbol{x}^R_{k,n+1}$ in place of $\boldsymbol{x}_{k,n+1}$. The iteration is complete.

b) In case $f\left(\boldsymbol{x}^R_{k,n+1}\right) \ge f\left(\boldsymbol{x}_{k,n}\right)$, a contraction is performed.

Contraction (Figures 3.4c–d)
The contraction is between $\bar{\boldsymbol{x}}_k$ and the better of $\boldsymbol{x}^R_{k,n+1}$ and $\boldsymbol{x}_{k,n+1}$. If $f\left(\boldsymbol{x}^R_{k,n+1}\right) \ge f\left(\boldsymbol{x}_{k,n+1}\right)$, the contraction is inside (Figure 3.4c). The new point is obtained as:

$$\boldsymbol{x}^{CI}_{k,n+1} = \bar{\boldsymbol{x}}_k - \gamma\left(\bar{\boldsymbol{x}}_k - \boldsymbol{x}_{k,n+1}\right),\ \gamma \in \mathbb{R}^+ \tag{3.5}$$

If $f\left(\boldsymbol{x}^{CI}_{k,n+1}\right) < f\left(\boldsymbol{x}_{k,n+1}\right)$, the contraction is successful and accept $\boldsymbol{x}^{CI}_{k,n+1}$ in place of $\boldsymbol{x}_{k,n+1}$. The iteration is complete. Otherwise go to the next step of shrinkage operation.

In case $f\left(\boldsymbol{x}_{k,n}\right) \le f\left(\boldsymbol{x}^R_{k,n+1}\right) < f\left(\boldsymbol{x}_{k,n+1}\right)$, the contraction is outside (Figure 3.4d). That is, the new point is:

$$x^{CO}_{k,n+1} = \bar{x}_k + \gamma\left(x^R_{k,n+1} - \bar{x}_k\right)x,\ \gamma \in \mathbb{R}^+ \tag{3.6}$$

If $f\left(\boldsymbol{x}^{CO}_{k,n+1}\right) < f\left(\boldsymbol{x}^R_{k,n+1}\right)$, accept $\boldsymbol{x}^{CO}_{k,n+1}$ in place of $\boldsymbol{x}_{k,n+1}$. The iteration is complete. Otherwise go to the next step of shrinkage operation.

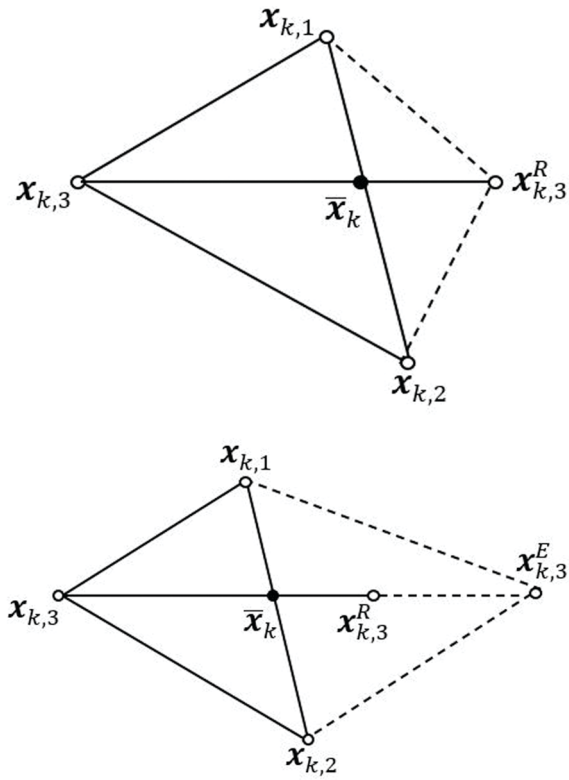


FIGURE 3.4a–b NM method – possible operations at the k^{th} iteration on a simplex in the two-dimensional case: (a) reflection, (b) expansion; $\bar{x}_k$ – centroid of the simplex.

Shrinkage (Figure 3.4e)
Shrink the n vertices $x_{k,2}, x_{k,3} \ldots, x_{k,n+1}$ towards the best point $x_{k,1}$ (Figure 3.4e) according to:

$$x^S_{k,i} = x_{k,1} + \sigma\left(x^S_{k,i} - x_{k,1}\right),\ i = 2,3,\ldots,n+1 \text{ and } \sigma \in \mathbb{R}^+ \tag{3.7}$$

With the new vertices $x_{k,1}, x^S_{k,2}, \ldots, x^S_{k,n+1}$ forming a new simplex, we go to the next iteration.

The termination criteria for the NM algorithm may be (a) the largest difference between adjacent vertices being less than a specified value ε_s or (b) the difference between the best and the worst solutions being less than a specified value ε_f. It is apparent that the convergence of the method depends on the parameters α, β, γ and σ. The standard values are $\alpha = 1, \beta = 2, \gamma = \frac{1}{2}$ and $\sigma = \frac{1}{2}$. As new vertices emerge during the iterations, the possibility of equal function values may surface. Following tie-breaking rules, a new vertex is indexed consistent with the relation

$$f\left(x_{k,1}\right) \le f\left(x_{k,2}\right) \le \ldots \le f\left(x_{k,n+1}\right) \tag{3.8}$$

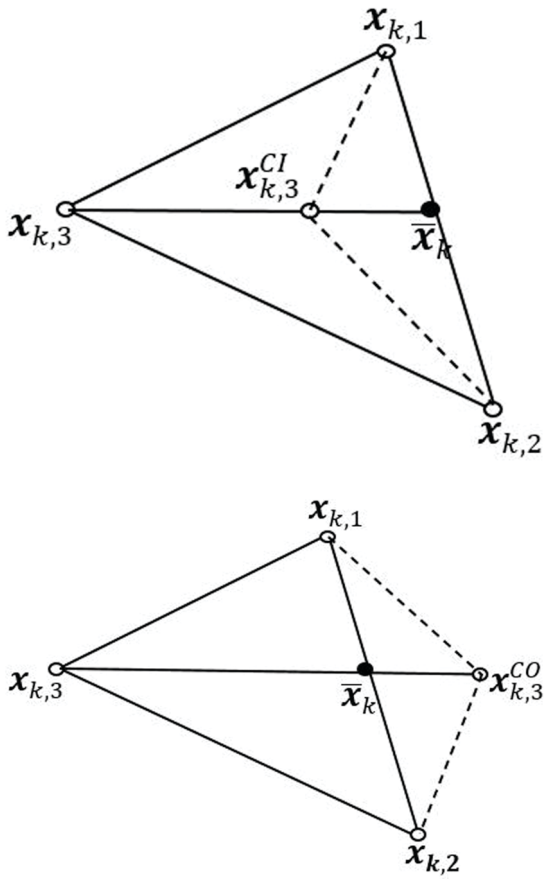


FIGURE 3.4c–d NM method – possible operations at the k^{th} iteration on a simplex in the two-dimensional case: (c) contraction inside, (d) contraction outside; $\bar{x}_k$ – centroid of the simplex.

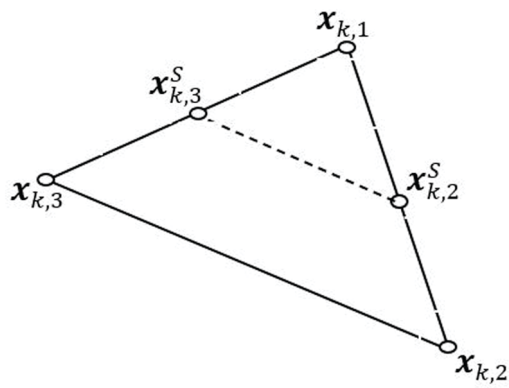


FIGURE 3.4e NM method – possible operations at the k^{th} iteration on a simplex in the two-dimensional case: (e) shrinkage; $\bar{x}_k$ – centroid of the simplex.

The convergence properties of the NM method are studied by Lagarias et al. (1998). The method is known to have the tendency of oscillations around a minimizer and is shown to converge to a non-minimizer (McKinnon 1998) even for convex functions of low dimension. Modifications to NM method for a possible improvement in convergence may be found in Kelley (1999), Tseng (1999) and Conn et al. (2009). Alternative variants may also be found in Byatt (2000) and Pham (2012).

Example 3.2. We solve by HJ and NM methods the maximum likelihood estimation (MLE) problem (Lehmann and Cassella 1998 and Papoulis 1991): fit a probability distribution for a given set of observed data.

Solution. The problem comes under the category of statistical inference or estimation. This is in contrast to the straightforward problem of statistical prediction about future observations from a probability distribution with known parameters. The statistical estimation is an inverse problem of estimating the parameters of a given distribution using an observed data. Suppose that the observed data is $\boldsymbol{z} = \{z_1, z_2, \ldots, z_n\}^T$. The observations are assumed to be realizations of random variables (RVs) $Z_i, i = 1, 2, \ldots, n$ which may follow the given probability distribution $\mathbb{F}_{\boldsymbol{Z}}(\boldsymbol{z}; \boldsymbol{\theta})$ provided that the parameters are suitably estimated. Here $\boldsymbol{Z}$ is the vector of the RVs $Z_i, i = 1, 2, \ldots, n$. $\boldsymbol{\theta}$, possibly a vector valued parameter, is unknown and needs to be estimated. $\mathbb{F}_Z(z; \theta)$ here is known as the joint CDF* of Z. Let the corresponding joint *pdf*† be $\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z}; \boldsymbol{\theta})$. In this context, $\mathbb{f}_{Z_i}(z_i; \boldsymbol{\theta})$ for each i is known as the marginal *pdf*‡ of

---

* Joint CDF

Consider a vector random variable $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$

The joint CDF, $\mathbb{F}_X(\boldsymbol{x}): \mathbb{R}^n \to [0,1]$ is defined as:

$$\mathbb{F}_X(\boldsymbol{x}) = P(X_1 \le x_1, X_2 \le x_2, \ldots, X_n \le x_n) \tag{i}$$

† Joint *pdf*

If $\mathbb{F}_X(\boldsymbol{x})$ is sufficiently differentiable, we define the joint *pdf* as $\dfrac{\partial^n \mathbb{F}_X(\boldsymbol{x})}{\partial x_1 \partial x_2 \ldots \partial x_n}$. Alternatively, $\mathbb{F}_X(\boldsymbol{x})$ is given by the n-dimensional integral in terms of the *pdf*:

$$\mathbb{F}_X(\boldsymbol{x}) = \int_{-\infty}^{x_1} \int_{-\infty}^{x_2} \ldots \int_{-\infty}^{x_n} f_X(x_1, x_2, \ldots, x_n)\, dx_1 dx_2 \ldots dx_n \tag{ii}$$

‡ Marginal *pdf*

The marginal density functions $\mathbb{f}_{X_k}(x_k), k = 1,2 \ldots, n$ follow from the existence of $\dfrac{\partial \mathbb{F}_{X_k}(x_k)}{\partial x_k}$ and are given by the following $(n-1)$ -dimensional integral with respect to dx_i, $i = 1, 2, \ldots n, i \ne k$:

$$\mathbb{f}_{X_k}(x_k) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} \mathbb{f}_{\boldsymbol{X}}(x_1, \ldots x_k \ldots, x_n) dx_1 \ldots dx_{k-1} dx_{k+1} \ldots \ldots dx_n \tag{iii}$$

each RV Z_i. As described in Appendix 1, the RVs are denoted by uppercase letters and their realizations by lowercase letters.

In estimating the parameter $\boldsymbol{\theta} \in \mathbb{R}^m$ by MLE, we form a likelihood function $L(\boldsymbol{\theta}; \boldsymbol{z}) : \mathbb{R}^m \to \mathbb{R}$ which is the joint *pdf* $\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z}; \boldsymbol{\theta})$ itself but now considered as a function of $\boldsymbol{\theta}$. The word 'likelihood' was used for the first time by Fisher (1922). Thus:

$$L(\boldsymbol{\theta}; \boldsymbol{z}) \triangleq \mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z}; \boldsymbol{\theta}) \tag{3.9}$$

The objective of the MLE is to find an estimate for $\boldsymbol{\theta}$ that maximizes $L(\boldsymbol{\theta}; \boldsymbol{z})$ with respect to $\boldsymbol{\theta}$. The estimate, say $\hat{\boldsymbol{\theta}}$, is meant to ensure that the observed data $\boldsymbol{z}$ is most likely to have been realized from the assumed pdf. Thus, the MLE gives the supremum of the likelihood function; i.e.:

$$\hat{\boldsymbol{\theta}} = \sup_{\theta} L(\boldsymbol{\theta}; \boldsymbol{z}) \tag{3.10}$$

Note that the maximization can be performed either on $L(\boldsymbol{\theta}; \boldsymbol{z})$ or on the log-likelihood function:

$$l(\boldsymbol{\theta}; \boldsymbol{z}) = log\ L(\boldsymbol{\theta}; \boldsymbol{z}) = log\ \mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z}; \boldsymbol{\theta}) \tag{3.11}$$

If $L(\boldsymbol{\theta}; \boldsymbol{z})$ is differentiable so the maximum exists, the familiar KKT condition applies, i.e.:

$$\frac{\partial L(\theta; z)}{\partial \theta} = 0 \text{ or } \frac{\partial l(\theta; z)}{\partial \theta} = 0 \tag{3.12}$$

Here $\frac{\partial}{\partial \theta} = \left(\frac{\partial}{\partial \theta_1}, \frac{\partial}{\partial \theta_2}, \ldots, \frac{\partial}{\partial \theta_m} \right)^T$. For the sake of this example, the marginal *pdf* $\mathbb{f}_{Z_i}(\boldsymbol{z}_i; \boldsymbol{\theta})$ is taken to be the generalized exponential pdf (Gupta and Kundu 2003) with $\boldsymbol{\theta}$ consisting of two scalar parameters α and λ:

$$\begin{aligned} \mathbb{f}_{Z_i}(z_i; \boldsymbol{\theta}) &= \alpha\lambda e^{-\lambda z_i}(1 - e^{-\lambda z_i})^{\alpha - 1}, \qquad z_i > 0 \\ &= 0, \text{ otherwise} \end{aligned} \tag{3.13}$$

We further assume that Z_i are independent and identically distributed (iid) RVs (see Appendix 1 for a definition of independent random variables) so that $\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z}; \boldsymbol{\theta}) = \prod_{i=1}^{n} \mathbb{f}_{Z_i}(z_i; \boldsymbol{\theta})$ and therefore:

$$L(\boldsymbol{\theta}; \boldsymbol{z}) = \prod_{i=1}^{n} \mathbb{f}_{Z_i}(z_i; \boldsymbol{\theta}) \tag{3.14a}$$

and

$$l(\boldsymbol{\theta}; z) = \sum_{i=1}^{n} log\ \mathbb{f}_{Z_i}(z_i; \boldsymbol{\theta}) \tag{3.14b}$$

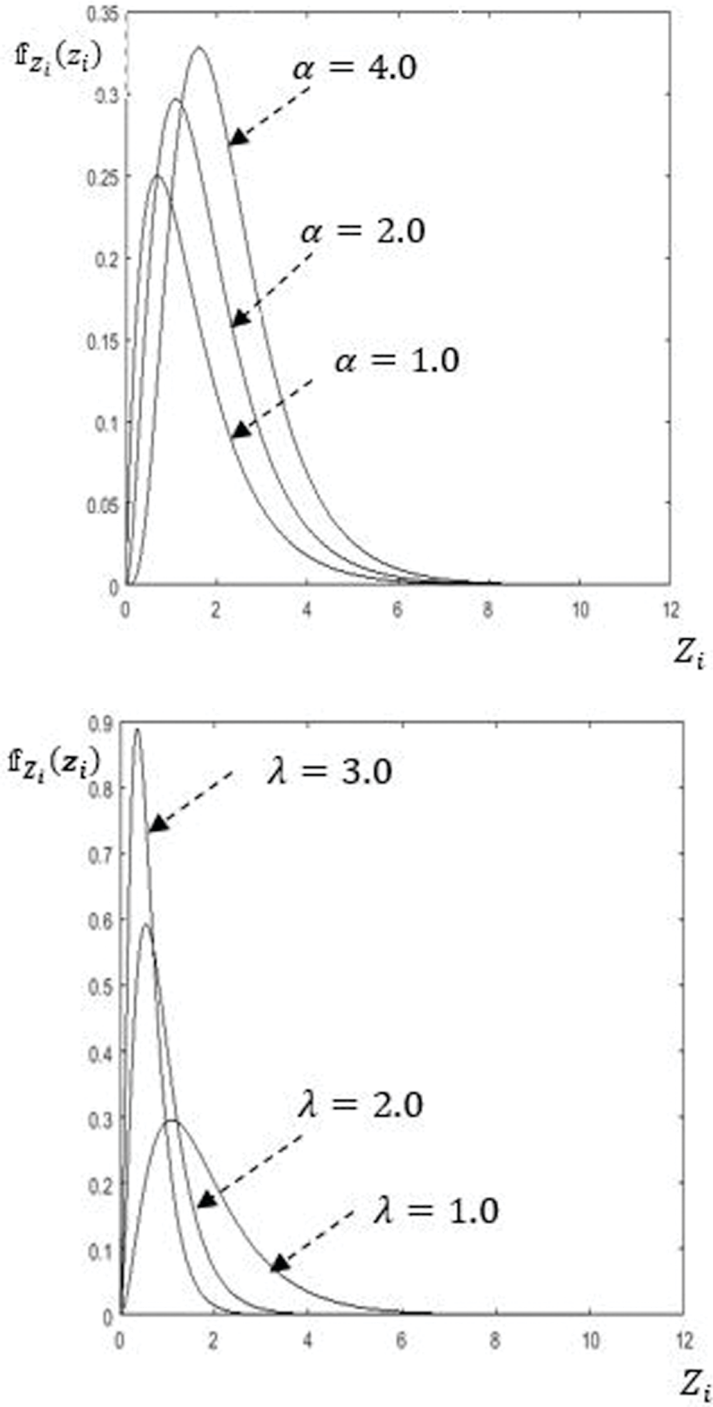


FIGURE 3.5a–b Generalized exponential pdf with different values of the two parameters α and λ: (a) α = 1,2,4 with λ = 1.0 and (b) λ = 1,2,3 with α = 2.0.

We need to estimate the vector $\boldsymbol{\theta}=(\alpha,\lambda)^T$, which we will take up shortly. The density function is shown in Figure 3.5 for different values of the parameters α and λ which are respectively called the shape and scale parameters. The pdf is often used to model the lifetime of a component or a system. The data z which is supposed to be available from actual observations, is presently simulated from the assumed pdf with reference values of $\alpha = 3.639$ and $\lambda = 2.239$. This exercise requires generating samples of z numerically and is related to the well-known problem of statistical prediction / sampling. See Appendix 3 for details on a few Monte Carlo (MC) simulation methods to realize samples of RVs from a specified distribution via transformation of RVs. These direct MC simulation methods are, in general, feasible for low dimensional sampling problems and one may need more efficient algorithms such as Markov chain Monte Carlo (MCMC) methods in the case of higher dimensional sampling. See Appendix 3 for a detailed description of Markov chains and MCMC sampling techniques. In the present example, the assumed pdf is one-dimensional and it suffices to use sampling by the MC method to get the required data z. With this available data, we use the MLE to yield the estimates $\hat{\alpha}$ and $\hat{\lambda}$ of the two parameters α and λ in the distribution.

Solution.
Using the independence property of the RVs Z_i, the joint *pdf* of Z is:

$$\mathbb{f}_Z(\boldsymbol{z};\boldsymbol{\theta}) = \prod_{i=1}^{n} \mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta}) = (\alpha\lambda)^n \prod_{i=1}^{n} e^{-\lambda z_i}(1-e^{-\lambda z_i})^{(\alpha-1)}, \qquad z_i > 0$$
$$= 0, \text{ otherwise} \tag{3.15}$$

In the present example, the optimization problem by MLE is only 2-dimensional and the estimates are obtainable by applying the necessary KKT conditions as expressed in Equation (3.12) (see also a similar Exercise 1.7 of Chapter 1). Here, we demonstrate the use of the derivative-free HJ and NM methods to get the solution. As a procedural convention followed so far, the estimates are obtained by posing the MLE as a minimization problem. That is, we minimize $-l(\boldsymbol{\theta};z)$. The optimization is carried out with the constraint $\alpha,\lambda > 0$. The results by HJ and NM methods are shown in Figures 3.6a–c and 3.6d–f, respectively, in the form of evolutions of the estimated parameters $\hat{\alpha}$ and $\hat{\lambda}$ during the iteration process with $n = 5000$. The same data z is used in obtaining the results by the two methods.

■

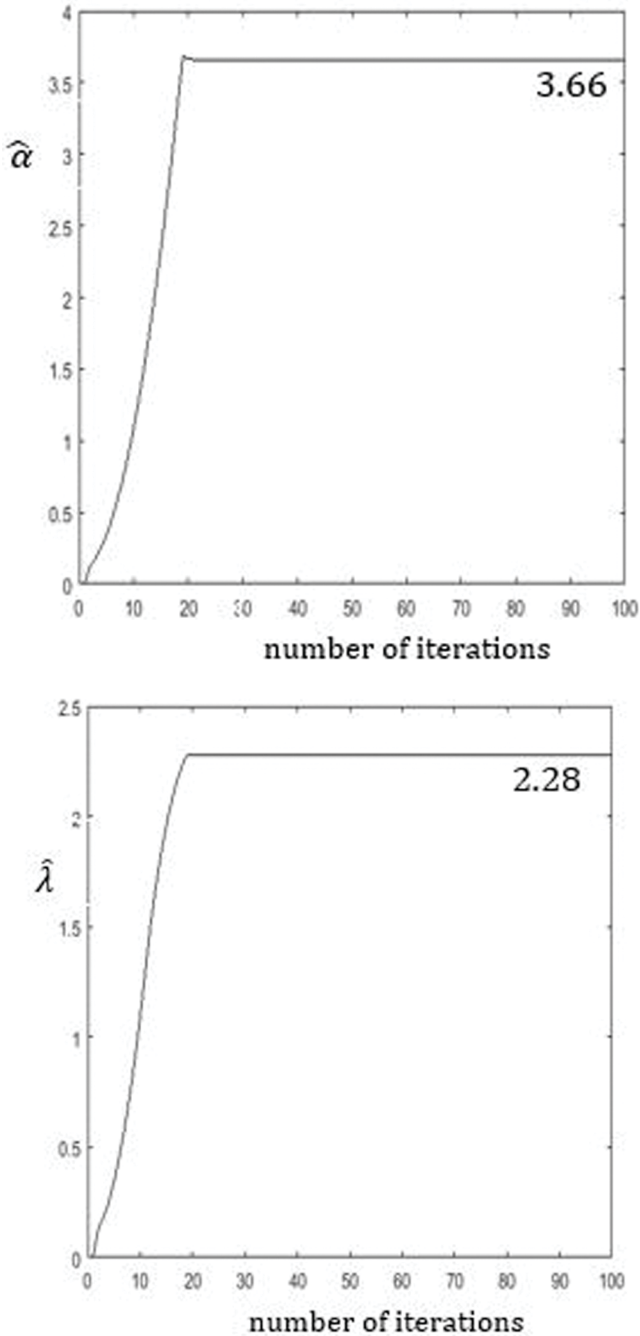


FIGURE 3.6a–b HJ method; statistical estimation by MLE of parameters of an assumed pdf using data of size $n = 5000$: (a) evolution of $\hat{\alpha}$ with iterations, (b) evolution of $\hat{\lambda}$ with iterations.

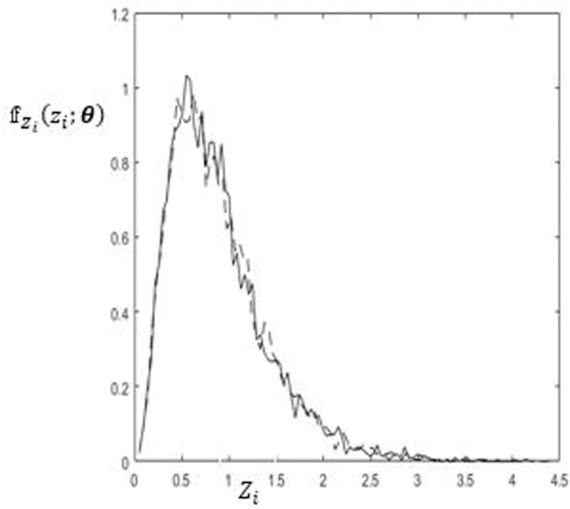


FIGURE 3.6c HJ method; statistical estimation by MLE of the parameters of an assumed pdf using data of size $n = 5000$; simulated *pdfs* with reference (true) and estimated parameters.

The curves shown in Figures 3.6e and 3.6f are simulated pdfs with data size $n = 5000$. Contrast the non-smoothness in the plotted graphs (with a finite data size) with the smooth theoretical pdf curves in Figure 3.5 which are drawn directly using Equation (3.13) with the reference parameters α and λ. For the theoretical pdf, Z_i is varied over [0,6.0] in steps of 0.1. On the other hand, the simulated pdfs in Figures 3.6e and 3.6f are histograms generated using the available data $\boldsymbol{z} \in \mathbb{R}^n$. The curves in dark line correspond to the reference parameters α and λ and the ones in dashed line to the MLE estimates $\hat{\alpha}$ and $\hat{\lambda}$. Note that $\boldsymbol{z}$ is simulated from Equation (3.13) and hence is the raw data. It is properly sorted before drawing the histograms.

Fisher information matrix and effectiveness of MLE

The MLE estimate as obtained above pertains to the observed data $\boldsymbol{z}$ which are given and hence fixed. Since $\boldsymbol{z}$ is assumed to be realized from $\mathbb{F}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})$, one indeed needs to consider the estimate $\hat{\boldsymbol{\theta}}$ as a RV and a function of $\boldsymbol{Z}$. As such, the likelihood function $L(\boldsymbol{\theta};\boldsymbol{Z})$ and the log-likelihood function $l(\boldsymbol{\theta};\boldsymbol{Z})$ are random vectors. Thus, by having different realizations of $\boldsymbol{z}$, one may have different estimates by MLE and obtain an approximate sampling distribution of $\hat{\boldsymbol{\theta}}$.

MLE is known to have the desirable large sample (asymptotic) property (Papoulis 1991). That is, for large n, $\hat{\boldsymbol{\theta}}$ approaches a normal distribution $N(\boldsymbol{\theta}, \boldsymbol{I}^{-1/2})$: a proof is provided in Appendix 3. Here $\boldsymbol{\theta}$ is the true value of the parameter set and $\boldsymbol{I} \in \mathbb{R}^{m\times m}$ is known as the Fisher information matrix (Fisher 1922, Lehmann and Cassella 1998). The matrix is defined by:

$$\boldsymbol{I}(\boldsymbol{\theta}) = E_{\boldsymbol{Z}}\left[\left(\frac{\partial l(\boldsymbol{\theta};\boldsymbol{Z})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial l(\boldsymbol{\theta};\boldsymbol{Z})}{\partial \boldsymbol{\theta}}\right)^T\right] = \int_{\mathbb{R}^m}\left(\frac{\partial l(\boldsymbol{\theta};\boldsymbol{Z})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial l(\boldsymbol{\theta};\boldsymbol{Z})}{\partial \boldsymbol{\theta}}\right)^T \mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})d\boldsymbol{z} \qquad (3.16)$$

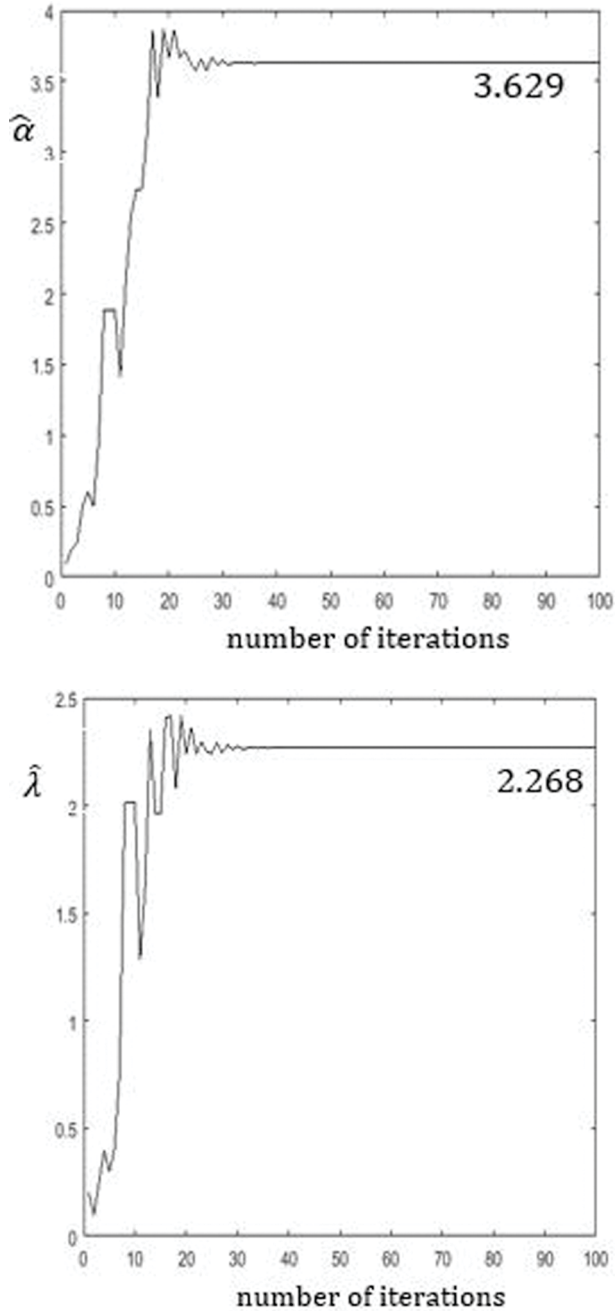


FIGURE 3.6d–e NM method; statistical estimation by MLE of the parameters of an assumed pdf using data of size $n = 5000$: (d) evolution of $\hat{\alpha}$ with iterations, (e) evolution of $\hat{\lambda}$ with iterations.

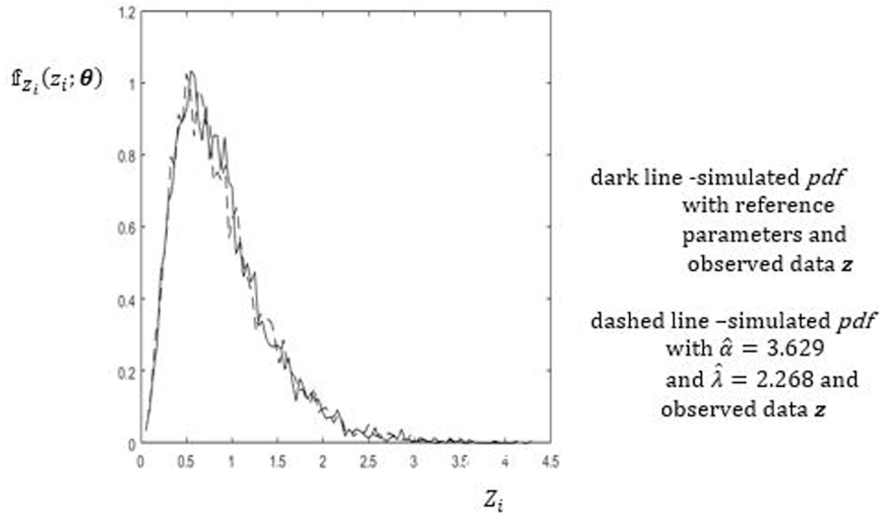


FIGURE 3.6f NM method; statistical estimation by MLE of the parameters of an assumed pdf using data of size $n = 5000$, simulated *pdfs* with reference (true) and estimated parameters.

The subscript $\boldsymbol{Z}$ in $E_{\boldsymbol{Z}}[.]$ denotes that the expectation is with respect to the probability measure corresponding to $\boldsymbol{Z}$; $d\mathbb{F}_{\boldsymbol{Z}} = \mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})d\boldsymbol{z}$. The derivative $\dfrac{\partial l(\boldsymbol{\theta};\boldsymbol{Z})}{\partial \boldsymbol{\theta}} \in \mathbb{R}^m$ is known as the score function denoted by $s(\boldsymbol{\theta};\boldsymbol{Z})$. Using Equation (3.16), we may derive FIM also as $\boldsymbol{I}(\boldsymbol{\theta}) = -E_{\boldsymbol{Z}}\left[\boldsymbol{H}(\boldsymbol{\theta})\right]$, where $\boldsymbol{H}(\boldsymbol{\theta})$ denotes the Hessian matrix. To do this, it is first necessary to show that $E_{\boldsymbol{Z}}\left[\dfrac{\partial l(\boldsymbol{\theta};\boldsymbol{Z})}{\partial \boldsymbol{\theta}}\right] = E_{\boldsymbol{Z}}\left[s(\boldsymbol{\theta};\boldsymbol{Z})\right] = 0$.

Towards this, let $\nabla = \dfrac{\partial}{\partial \boldsymbol{\theta}}$. Then:

$$E_{\boldsymbol{Z}}\left[s(\boldsymbol{\theta};\boldsymbol{Z})\right] = E_{\boldsymbol{Z}}\left[\nabla \log L(\boldsymbol{\theta};\boldsymbol{Z})\right]$$

$$= \sum_{i=1}^{n} \int_{\mathbb{R}} \frac{\nabla\, \mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})}{\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})} \mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta}) dz_i$$

$$= \sum_{i=1}^{n} \int_{\mathbb{R}} \nabla\, \mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})\, dz_i = \sum_{i=1}^{n} \nabla \int_{\mathbb{R}} \mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})\, dz_i = 0 \qquad (3.17)$$

since each integrand in the last equation is unity. Now we prove that $\boldsymbol{I}(\boldsymbol{\theta}) = -E_Z\left[\boldsymbol{H}(\boldsymbol{\theta})\right]$.

One has: $\nabla^2 = \left[\dfrac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T}\right]_{m\times m}$, i.e.:

$$\nabla^2 = \begin{bmatrix} \dfrac{\partial^2}{\partial\theta_1^2} & \dfrac{\partial^2}{\partial\theta_1\partial\theta_2} & \cdots & \dfrac{\partial^2}{\partial\theta_1\partial\theta_m} \\ \dfrac{\partial^2}{\partial\theta_2\partial\theta_1} & \dfrac{\partial^2}{\partial\theta_2^2} & \cdots & \dfrac{\partial^2}{\partial\theta_2\partial\theta_m} \\ & & \cdots & \\ & & \cdots & \\ \dfrac{\partial^2}{\partial\theta_m\partial\theta_1} & \dfrac{\partial^2}{\partial\theta_m\partial\theta_2} & \cdots & \cdots\dfrac{\partial^2}{\partial\theta_m^2} \end{bmatrix} \tag{3.18}$$

$\boldsymbol{H}(\boldsymbol{\theta})$ is given by:

$$\begin{aligned} \boldsymbol{H}(\boldsymbol{\theta}) &= \nabla^2 l(\boldsymbol{\theta};\boldsymbol{Z}) = \nabla s(\boldsymbol{\theta};\boldsymbol{Z}) \\ &= \nabla\left(\sum_{i=1}^{n}\frac{\nabla\, \mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})}{\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})}\right) \\ &= \sum_{i=1}^{n}\left\{\frac{\nabla^2\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})}{\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})} - \frac{1}{\left(\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})\right)^2}\nabla\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})\,\nabla\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})^T\right\} \\ &= \sum_{i=1}^{n}\frac{\nabla^2\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})}{\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})} - s(\boldsymbol{\theta};\boldsymbol{Z})s(\boldsymbol{\theta};\boldsymbol{Z})^T \end{aligned} \tag{3.19}$$

The expectation of $\boldsymbol{H}(\boldsymbol{\theta})$ is:

$$E_{\boldsymbol{Z}}[H(\boldsymbol{\theta})] = E_{\boldsymbol{Z}}\left[\sum_{i=1}^{n}\frac{\nabla^2\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})}{\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})}\right] - E_{\boldsymbol{Z}}[s(\boldsymbol{\theta};\boldsymbol{Z})s(\boldsymbol{\theta};\boldsymbol{Z})^T] \tag{3.20}$$

T is:

$$\begin{aligned} E_{\boldsymbol{Z}}\sum_{i=1}^{n}\frac{\nabla^2\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})}{\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})} &= \sum_{i=1}^{n}\int_{\mathbb{R}}\frac{\nabla^2\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})}{\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})}\,\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})\,dz_i \\ &= \sum_{i=1}^{n}\int_{\mathbb{R}}\nabla^2\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})\,dz_i = \sum_{i=1}^{n}\nabla^2\int_{\mathbb{R}}\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})\,dz_i \end{aligned} \tag{3.21a}$$

Therefore:

$$E_{\boldsymbol{Z}}\left[\boldsymbol{H}(\boldsymbol{\theta})\right] = -E_{\boldsymbol{Z}}\left[s(\boldsymbol{\theta};\boldsymbol{Z})s(\boldsymbol{\theta};\boldsymbol{Z})^T\right] \tag{3.21b}$$

Thus, from Equation (3.16) and the last equation, we get as $n \to \infty$:

$$\boldsymbol{I}(\boldsymbol{\theta}) = -E_{\boldsymbol{Z}}\left[\boldsymbol{H}(\boldsymbol{\theta})\right] \tag{3.22}$$

The $(p,q)^{th}$ element of this matrix is:

$$\begin{aligned}\left[\boldsymbol{I}\right]_{pq}(\boldsymbol{\theta}) &= E_Z\left[\left(\frac{\partial \log L((\boldsymbol{\theta};\boldsymbol{Z}))}{\partial \theta_p}\right)\left(\frac{\partial \log L(\boldsymbol{\theta};\boldsymbol{Z})}{\partial \theta_q}\right)\right] \\ &= E_Z\left[s_p(\boldsymbol{\theta};\boldsymbol{Z})s_q(\boldsymbol{\theta};\boldsymbol{Z})\right]\end{aligned} \tag{3.23}$$

where $\dfrac{\partial \log L(\boldsymbol{\theta};\boldsymbol{Z})}{\partial \theta_p} = s_p(\boldsymbol{\theta};\boldsymbol{Z})$, the p^{th} element of the score function, i.e. the first-order derivative of $\log L(\boldsymbol{\theta};\boldsymbol{Z})$ with respect to θ_p. $E_Z\left[s_p(\boldsymbol{\theta};\boldsymbol{Z})s_q(\boldsymbol{\theta};\boldsymbol{Z})\right]$, which is also denoted as $cov\left(s_p(\boldsymbol{\theta};\boldsymbol{Z})s_q(\boldsymbol{\theta};\boldsymbol{Z})\right)$, stands for the covariance (see covariance matrix definition in Equation A1.22 of Appendix 1) of the random variables $s_p(\boldsymbol{\theta};\boldsymbol{Z})$ and $s_q(\boldsymbol{\theta};\boldsymbol{Z})$. Having proved $\boldsymbol{I}(\boldsymbol{\theta}) = -E_{\boldsymbol{Z}}\left[\boldsymbol{H}(\boldsymbol{\theta})\right]$, we may now infer that it quantifies the overall sensitivity of $\mathbb{F}_{\boldsymbol{Z}}$ with respect to $\boldsymbol{\theta}$ via an averaging process. With $\boldsymbol{I}(\boldsymbol{\theta})$ determined from Equation (3.16), one has the sampling distribution of $\hat{\boldsymbol{\theta}}$ as $N\left(\boldsymbol{\theta}, \boldsymbol{I}^{-1/2}\right)$. Thus, for large *n*, the mean vector $E\left[\hat{\boldsymbol{\theta}}\right] = \boldsymbol{\theta}$ with the covariance matrix being $\boldsymbol{I}^{-1}$. This shows that MLE is an unbiased estimator in that $E\left[\hat{\boldsymbol{\theta}}\right] \to \boldsymbol{\theta}$ for large *n*. It is usually a difficult task to obtain $I(\boldsymbol{\theta})$ computationally. However, for large sample sizes, one uses $\boldsymbol{H}\left(\hat{\boldsymbol{\theta}}\right)$ replacing $I(\boldsymbol{\theta})$. Thus $\boldsymbol{H}\left(\hat{\boldsymbol{\theta}}\right)$ is often called the observed FIM.

When using a large sample, FIM may also be used to obtain approximate confidence intervals (Appendix 3) for the estimated parameter. In other words, the mean square error in the estimation process for each element $\hat{\theta}_i$ is obtainable from FIM. For the present example, the observed FIM by the HJ method is obtained as: $\boldsymbol{H}\left(\hat{\boldsymbol{\theta}}\right) = \begin{bmatrix} 375.3 & -833.0 \\ -833.0 & 2834.3 \end{bmatrix}$. A finite difference scheme is utilized in computing the Hessian matrix $\boldsymbol{H}(\boldsymbol{\theta}) = \nabla^2 \log L(\boldsymbol{\theta};\boldsymbol{Z})$ at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$. Since $\hat{\boldsymbol{\theta}}$ approaches $N\left(\boldsymbol{\theta}, \boldsymbol{I}^{-1/2}\right)$ as $n \to \infty$, the variances of $\hat{\alpha}$ and $\hat{\lambda}$ are given by the diagonal terms of $\left[\boldsymbol{H}\left(\hat{\boldsymbol{\theta}}\right)\right]^{-1}$: $\sigma^2_{\hat{\alpha}} = 0.0077$ and $\sigma^2_{\hat{\lambda}} = 0.001$. The 95% confidence intervals are $\alpha \pm 1.96\ \sigma_{\hat{\alpha}} = (3.467, 3.811)$ for α and $\lambda \pm 1.96\ \sigma_{\hat{\lambda}} = (2.177, 2.301)$ for λ. α and λ are the mean value of the RVs $\hat{\alpha}$ and $\hat{\lambda}$ corresponding to their asymptotic distributions. These are respectively

taken as 3.639 and 2.239, the reference values. $\boldsymbol{H}\left(\hat{\boldsymbol{\theta}}\right)$ for the NM method is $\begin{bmatrix} 394.7 & -864.6 \\ -864.6 & 2901.0 \end{bmatrix}$ and the 95% confidence intervals for the estimates obtained are (3.472, 3.806) for α and (2.177, 2.301) for λ.

3.3 OTHER DIRECT SEARCH METHODS

Rosenbrock's rotating coordinates method (1960) and Powell's conjugate directions method (1964) are a few other direct search methods based on simple and interesting heuristics. Both the methods are marked by some unique features that distinguish them from the rest.

3.3.1 Rotating Coordinates Method of Rosenbrock

Starting with an initial $\boldsymbol{x}_0 \in \mathbb{R}^n$ and step sizes $s_i^k, i = 1,2,\ldots,n$, the method starts with the n Euclidean axes as search directions to locate an improved $\boldsymbol{x} \in \mathbb{R}^n$ as in HJ method. The search continues over the n directions cyclically until every direction returns at least one success and one failure. This marks the end of a stage. The next stage initiates search along a new set of n orthogonal directions which are generated using Gram-Schmidt orthogonalization procedure (Appendix 3). This is the distinctive feature of the method. Details of the computational procedure at any k^{th} stage are given below.

With $\boldsymbol{x}^{k-1}$ being the solution at the end of $(k-1)^{th}$ stage, let the k^{th} stage start at the location $\boldsymbol{x}_0^k = \boldsymbol{x}^{k-1}$. Let $\boldsymbol{d}_i^k, i = 1,2,\ldots,n$ be a set of unit orthogonal vectors. From $\boldsymbol{x}^{k-1}$, the search begins in the direction $\boldsymbol{d}_1^k$ with the step size s_1^k. If $f\left(\boldsymbol{x}^{k-1} + s_1^k \boldsymbol{d}_1^k\right) \le f\left(\boldsymbol{x}^{k-1}\right)$, the step is deemed to be a success and $\boldsymbol{x}^k = \boldsymbol{x}^{k-1} + s_1^k \boldsymbol{d}_1^k$. The step size s_1^k is increased to βs_1^k with $\beta > 1$ to be used in the next cycle. In case $f\left(\boldsymbol{x}^{k-1} + s_1^k \boldsymbol{d}_1^k\right) > f\left(\boldsymbol{x}^{k-1}\right)$, the step is a failure and $\boldsymbol{x}^{k-1}$ is not updated and s_1^k is reduced to γs_1^k with $\gamma < 1$ before proceeding to the next direction $\boldsymbol{d}_2^k$. The search direction $\boldsymbol{d}_2^k$ is perturbed with the step size s_2^k. Rosenbrock recommends the values of 3.0 and 0.5 for β and γ respectively. The search along an i^{th} direction $\boldsymbol{d}_i^k$ with a failure in the earlier cycle is performed in the subsequent cycle in the opposite direction with the modified step size γs_1^k. The cyclic searches are continued until there is one success and one failure in every direction, when the k^{th} stage is considered complete. Let $\boldsymbol{x}^k$ be the final solution at the end of this stage. Operations involved in the first stage are demonstrated in Table 3.1 which pertains to the example problem 3.3.

Generation of the new set of n orthogonal directions before $(k+1)^{th}$ stage starts

The first direction $\boldsymbol{d}_1^{k+1}$ for the next stage is chosen parallel to $\boldsymbol{x}^k - \boldsymbol{x}^{k-1} = \sum_i^n t_i^k \boldsymbol{d}_i^k$ where t_i^k is the algebraic sum of all the successful steps (net distance travelled) in each

direction $\boldsymbol{d}_i^k$ at the end of the k^{th} stage. The remaining directions $\boldsymbol{d}_i^{k+1}, i = 2,\ldots,n$ are chosen orthonormal to each other and to $\boldsymbol{d}_1^{k+1}$ by the Gram-Schmidt orthogonalization procedure. To this end, define n vectors $\boldsymbol{A}_i^k, i = 1,2,\ldots,n$:

$$\begin{aligned}
\boldsymbol{A}_1^k &= t_1^k \boldsymbol{d}_1^k + t_2^k \boldsymbol{d}_2^k + \ldots + t_n^k \boldsymbol{d}_n^k \\
\boldsymbol{A}_2^k &= t_2^k \boldsymbol{d}_2^k + \ldots + t_n^k \boldsymbol{d}_n^k \\
\boldsymbol{A}_n^k &= t_n^k \boldsymbol{d}_n^k
\end{aligned} \tag{3.24}$$

$\boldsymbol{A}_1^k$, normalized as shown below, is the new first direction:

$$\boldsymbol{d}_1^{k+1} = \frac{\boldsymbol{A}_1^k}{\left\|\boldsymbol{A}_1^k\right\|}$$

For $i = 2,3,\ldots,n$, the new directions are constructed by the Gram-Schmidt procedure as:

$$\boldsymbol{B}_i^k = \boldsymbol{A}_i^k - \sum_{j=1}^{i-1} \left(\left(\boldsymbol{A}_i^k\right)^T \boldsymbol{d}_j^{k+1} \right) \boldsymbol{d}_j^{k+1} \text{ and } \boldsymbol{d}_i^{k+1} = \frac{\boldsymbol{B}_i^k}{\left\|\boldsymbol{B}_i^k\right\|} \tag{3.25}$$

It may be verified that $\boldsymbol{d}_i^{k+1}$, $i = 2,3,\ldots,n$ are orthogonal to $\boldsymbol{d}_1^{k+1}$ and also to each other. Computations restart for this stage with the new search directions. The stopping criterion may be based on the convergence of the objective function.

Example 3.3. We solve the MLE problem of Example 3.2 by Rosenbrock's rotating coordinates method.

Solution. The starting point $\boldsymbol{x}_0$ is chosen as $(3,3)^T$. Computations performed in the first stage are shown in Table 3.1.

Notice that the first stage is considered complete when the search in the $\boldsymbol{d}_2^1$ recorded at least one success at the end of the sixth cycle while searches in the earlier cycles are all failures. The point at the end of the first stage is $\boldsymbol{x}^1 = (5.875, 2.969)^T$ which is the starting point for the next stage (Figure 3.7). The first stage is complete after having achieved a success in each of the two directions. The new directions for the next stage are obtained by the Gram-Schmidt procedure using Equations (3.24–3.25). The orthogonal directions generated for the first few stages are shown in Figure 3.7. The final result is shown in Figure 3.8. The result pertains to the same data $\boldsymbol{z}$ as was used by the HJ and NM methods.

TABLE 3.1
MLE Problem: Computational Steps in the First Stage of Rosenbrock's Rotating Coordinates Method; $n = 2$, $x_0 = (3, 3)^T$

$f(x_0) = 4145.51$, $\beta = 3$, $\gamma = 0.5$, Initial Search Vector $d^1 = \left(d_1^1, d_2^1\right) = \left[\begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}\right]$

j^{th} direction	s_1^1	s_2^1	t_1^1	t_2^1	Parameter $\alpha = x_1^1$	Parameter $\lambda = x_2^1$	$f(x^1)$	S: success; F: failure
d_1^1	1.0	1.0	1.0	0.0	4.0	3.0	3604.98	S
d_2^1	3.0	1.0	1.0	0.0	4.0	4.0	5579.02	F
d_1^1	3.0	–0.5	4.0	0.0	7.0	3.0	3500.53	S
d_2^1	9.0	-0.5	4.0	0.0	7.0	2.5	3900.25	F
d_1^1	9.0	0.25	4.0	0.0	16.0	3.0	7448.03	F
d_2^1	–4.5	0.25	4.0	0.0	7.0	3.25	3562.01	F
d_1^1	–4.5	–0.125	4.0	0.0	2.5	3.0	4608.18	F
d_2^1	2.25	–0.125	4.0	0.0	7.0	2.875	3527.87	F
d_1^1	2.25	0.0625	4.0	0.0	9.25	3.0	4127.19	F
d_2^1	–1.125	0.0625	4.0	0.0	7.0	3.0625	3502.15	F
d_1^1	–1.125	–0.031	2.2875	0.0	5.875	3.0	3366.44	S
d_2^1	–3.375	–0.031	2.2875	–0.031	5.875	2.969	3352.57	S

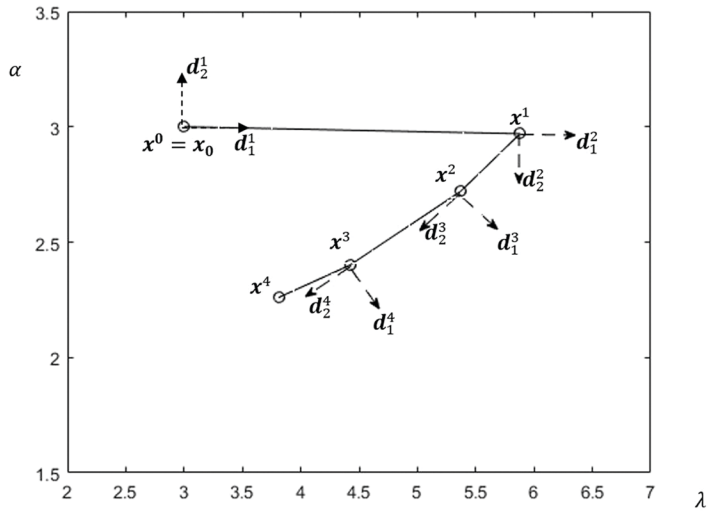


FIGURE 3.7 Rosenbrock's rotating coordinates method; solution to the MLE problem in Example 3.2 with $n = 2, \boldsymbol{d}^k, k > 1$ are generated by the Gram-Schimdt procedure at the beginning of each stage; at the initial stage $\boldsymbol{d}^1$ corresponds to the n Euclidean axes, $\boldsymbol{x}^k \left(k \geq 1\right)$ is the solution at the end of the k^{th} stage.

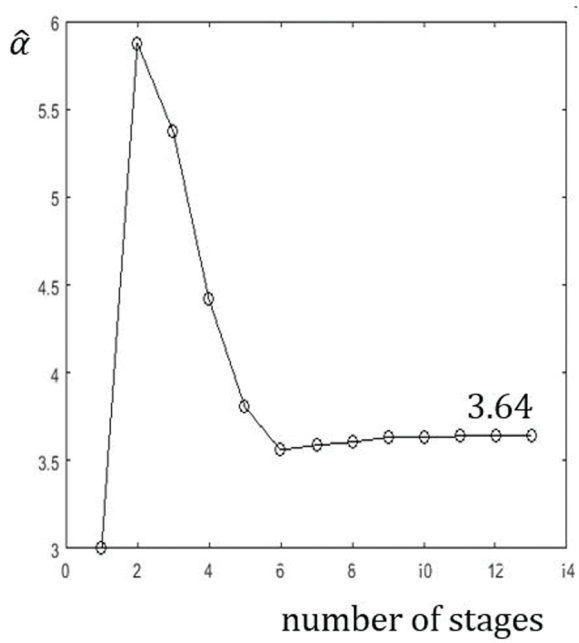


FIGURE 3.8a Rosenbrock's rotating coordinates method; solution to the MLE problem in Example 3.2: (a) evolution of the estimated parameter $\hat{\alpha}$ with stages and (b) evolution of the estimated parameter $\hat{\lambda}$ with stages; final solution: $\hat{\alpha} = 3.64$ and $\hat{\lambda} = 2.249$ (as against the reference values 3.639 and 2.239, respectively).

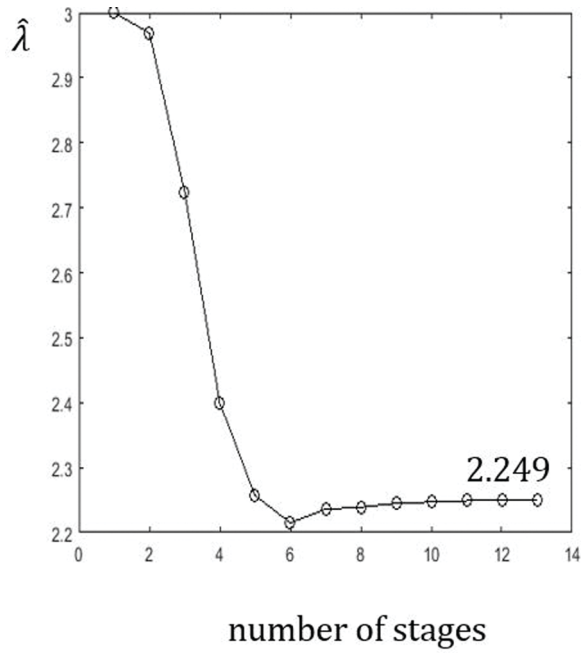


FIGURE 3.8b (Continued)

■

3.3.2 Powell's Method of Conjugate Directions

The method partly replicates, albeit with significant differences, the derivative-based CG method of Fletcher and Reeves (1964). The resemblance with the CG method is in the use of conjugate directions during the iteration process. Otherwise, the method is strikingly different in that the conjugate directions are generated only by direct searches without a computation of the derivatives. As is known, by using conjugate directions, an n-dimensional quadratic function is minimized in n-steps irrespective of the initial $\boldsymbol{x}_0$. Even otherwise, a non-quadratic function may often be well approximated by a quadratic function near its optimum so that a CG method accelerates the convergence and optimizes the function in a finite number of steps. If we consider a quadratic function $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{B}^T\boldsymbol{x} + c$ as in Equation (2.10) of Chapter 2, the optimum $\boldsymbol{x}^*$ may be expressed by Equation (2.15) which is restated below:

$$\boldsymbol{x}^* = \boldsymbol{x}_0 + \sum_{i=0}^{n-1} \bar{s}_i \bar{\boldsymbol{d}}_i \tag{3.26}$$

$\bar{s}_i, i = 0,1,\ldots,n-1$ are the unknown step sizes. $\bar{\boldsymbol{d}}_i$, $i = 0,1,\ldots n-1$, are $\boldsymbol{Q}$-conjugate directions, i.e.:

$$\bar{\boldsymbol{d}}_i^T \boldsymbol{Q} \bar{\boldsymbol{d}}_j = 0, i \neq j \tag{3.27}$$

In this method, determination of $\bar{\boldsymbol{d}}_i$ is based on the following parallel subspace property.

'Given a quadratic function $f(\boldsymbol{z}): \mathbb{R}^n \to \mathbb{R}$ with $\boldsymbol{z} \in \mathbb{R}^n$ and two parallel hyperplanes of dimension $m < n$, the line joining the stationary points $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ of $f(\boldsymbol{z})$ in the hyperplanes is conjugate to any line parallel to the hyperplanes'.

In Figure 3.9, the property is illustrated in a two-dimensional case. $\boldsymbol{z}_1$ is the minimum point of a quadratic function $f(\boldsymbol{z})$ starting from $\boldsymbol{z}_0^{(1)}$ along the direction $\boldsymbol{d}$ in the first hyperplane and $\boldsymbol{z}_2$ the minimum point starting from $\boldsymbol{z}_0^{(2)}$ along the direction $\boldsymbol{d}$ in the parallel hyperplane. Therefore, with $\nabla f(\boldsymbol{z}) = \boldsymbol{Q}\boldsymbol{z} + \boldsymbol{b}$:

$$\nabla f^T(\boldsymbol{z}_1)\boldsymbol{d} = 0 \Rightarrow (\boldsymbol{Q}\boldsymbol{z}_1 + \boldsymbol{B})^T \boldsymbol{d} = 0 \tag{3.28a}$$

and similarly:

$$\nabla f^T(\boldsymbol{z}_2)\boldsymbol{d} = 0 \Rightarrow (\boldsymbol{Q}\boldsymbol{z}_2 + \boldsymbol{B})^T \boldsymbol{d} = 0 \tag{3.28b}$$

It follows that:

$$0 = (\boldsymbol{Q}\boldsymbol{z}_2 + \boldsymbol{b})^T \boldsymbol{d} - (\boldsymbol{Q}\boldsymbol{z}_1 + \boldsymbol{b})^T$$

$$\Rightarrow (\boldsymbol{z}_2 - \boldsymbol{z}_1)^T \boldsymbol{Q}\boldsymbol{d} = 0 \tag{3.29}$$

Thus, if the new direction is chosen as $\bar{\boldsymbol{d}} = \boldsymbol{z}_2 - \boldsymbol{z}_1$, then $\bar{\boldsymbol{d}}$ is $\boldsymbol{Q}$-conjugate to $\boldsymbol{d}$.

The method utilizes the above property and generates the conjugate directions as iterations progress. An illustration is given in Figure 3.10 for the two-dimensional case. We initialize the search directions along the two coordinate directions $\boldsymbol{d}_1 = (1,0)^T$ and $\boldsymbol{d}_2 = (0,1)^T$ in the Euclidean space E^2. With starting point $\boldsymbol{x}_0$, line search is first undertaken along $\boldsymbol{d}_2$ with a step size s to find a minimum at $\boldsymbol{x}_1$ such that $s = \arg\min f(\boldsymbol{x}_0 + \hat{s}\boldsymbol{d}_2)$. Thus $\boldsymbol{x}_1 = \boldsymbol{x}_0 + s\boldsymbol{d}_2$. Minima $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ are similarly found along $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ respectively by line search. Note that in each cycle, an extra search is performed along the starting direction (here along $\boldsymbol{d}_2$).

As per the parallel subspace property, $\bar{\boldsymbol{d}}_1 = \boldsymbol{x}_3 - \boldsymbol{x}_1$ is $\boldsymbol{Q}$-conjugate to $\boldsymbol{d}_2$. This completes one cycle of iteration. In the second cycle, $\boldsymbol{d}_1$ is discarded. A start is made from $\boldsymbol{x}_3$ with line searches along $\bar{\boldsymbol{d}}_1$ and $\boldsymbol{d}_2$ to get the stationary points $\boldsymbol{x}_4$ and $\boldsymbol{x}_5$ respectively. As in the first cycle, an extra search along $\bar{\boldsymbol{d}}_1$ is performed to obtain $\boldsymbol{x}_6$. One gets the second conjugate direction $\bar{\boldsymbol{d}}_2 = \boldsymbol{x}_6 - \boldsymbol{x}_4$. It is easy to see that $\bar{\boldsymbol{d}}_2$ is $\boldsymbol{Q}$-conjugate to $\bar{\boldsymbol{d}}_1$ (because of the parallel subspace property). Let a line search along $\bar{\boldsymbol{d}}_2$ reach the stationary point $\boldsymbol{x}_7$. This completes one stage of iteration. Before proceeding to the next stage, we test the convergence by checking if $|f(\boldsymbol{x}_7) - f(\boldsymbol{x}_6)| \leq \varepsilon$, a

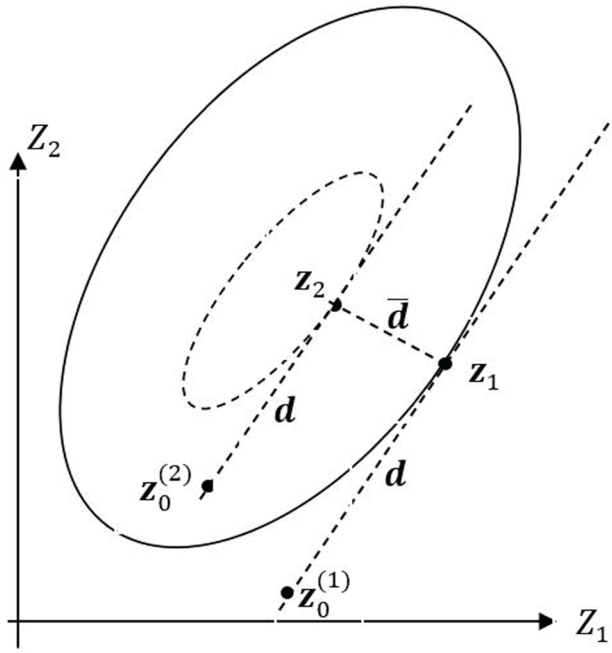


FIGURE 3.9 Powell's method of conjugate directions; determining a conjugate direction $\bar{\boldsymbol{d}}$ in a two-dimensional case.

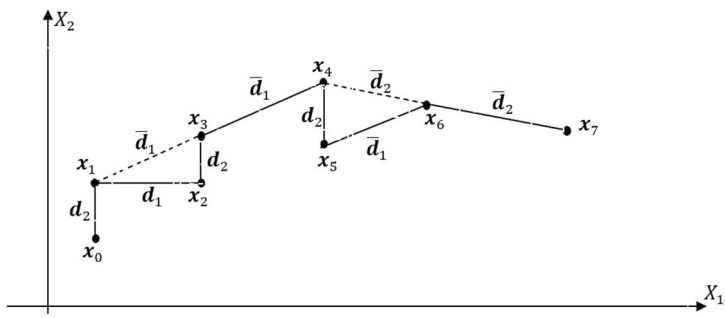


FIGURE 3.10 Powell's method of conjugate directions: two-dimensional case, the generated conjugate directions $\bar{\boldsymbol{d}}_1 = \boldsymbol{x}_3 - \boldsymbol{x}_1$ and $\bar{\boldsymbol{d}}_2 = \boldsymbol{x}_6 - \boldsymbol{x}_4$ at the end of the first stage consisting of two cycles of iteration.

specified small value to stop the iterations. If the convergence criterion is not satisfied, a new stage is started from the current point $\boldsymbol{x}_7$ by reinitializing $\boldsymbol{d}_i, i = 1,2$ along the two orthogonal coordinate directions in E_2. For a non-quadratic function $f(\boldsymbol{x})$, stages of iteration need be continued till convergence. Table 3.2 gives the details of the algorithm for a general *n*-dimensional case.

TABLE 3.2
Algorithm of Powell's Method of Conjugate Directions

Given an objective function $f(\boldsymbol{x})$ in $\mathbb{R}^n$ and a starting point $\boldsymbol{x}_0$, Powell's method consists of stages each of which has n cycles of iteration. At the k^{th} stage, the method may be described as follows.

Step 1. Proceed with the first cycle; initialize the search directions $\boldsymbol{d}_i, i = 1, 2, \ldots, n$ along the n coordinate directions in E_n.

Step 2. Starting from $\boldsymbol{x}_0$, perform an initial line search along $\boldsymbol{d}_n$ so as to reach $\boldsymbol{x}_1$ such that: $s = \arg\min f\left(\boldsymbol{x}_0 + \hat{s}\boldsymbol{d}_n\right)$. Therefore, $\boldsymbol{x}_1 = \boldsymbol{x}_0 + s\boldsymbol{d}_n$. Next, perform similar line searches along all $\boldsymbol{d}_i, i = 1, 2, \ldots, n$ (including $\boldsymbol{d}_n$ again) with optimum step sizes s_i so that $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + s_i\boldsymbol{d}_i$. $i = 1, 2, \ldots, n$.

Step 3. Set $\boldsymbol{d}_{n+1} = \boldsymbol{x}_{n+1} - \boldsymbol{x}_1$ which is Q-conjugate to $\boldsymbol{d}_n$ as per the parallel subspace property. This completes one cycle of iteration.

Step 4. If n cycles are not completed, discard $\boldsymbol{d}_1$ and let $\boldsymbol{d}_i = \boldsymbol{d}_{i+1}$, $i = 1, 2, \ldots, n$.

Set $\boldsymbol{x}_0 = \boldsymbol{x}_{n+1}$ and go to step 2.

Step 5. If n cycles are completed, it marks the end of a stage. Check for convergence, which, if achieved, marks the end of iterations. Otherwise go to step 6.

Step 6. Go to the next stage (i.e. to step 2) with the current point as $\boldsymbol{x}_0$ and after re-initializing the search directions $\boldsymbol{d}_i, i = 1, 2, \ldots, n$ along the n coordinate directions in E_n.

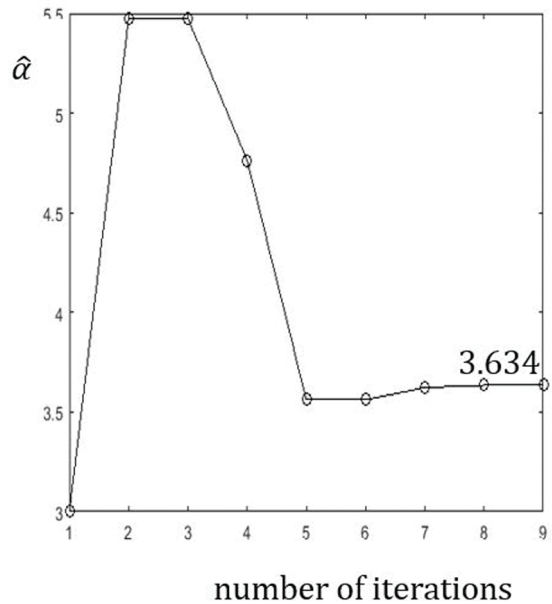


FIGURE 3.11a Powell's conjugate directions method and solution to the MLE problem in Example 3.2. (a) Evolution of the estimated parameter $\hat{\alpha}$ with iterations and (b) evolution of the estimated parameter $\hat{\lambda}$ with iterations; final solution: $\hat{\alpha} = 3.634$ and $\hat{\lambda} = 2.269$ (as against the reference values 3.639 and 2.239, respectively).

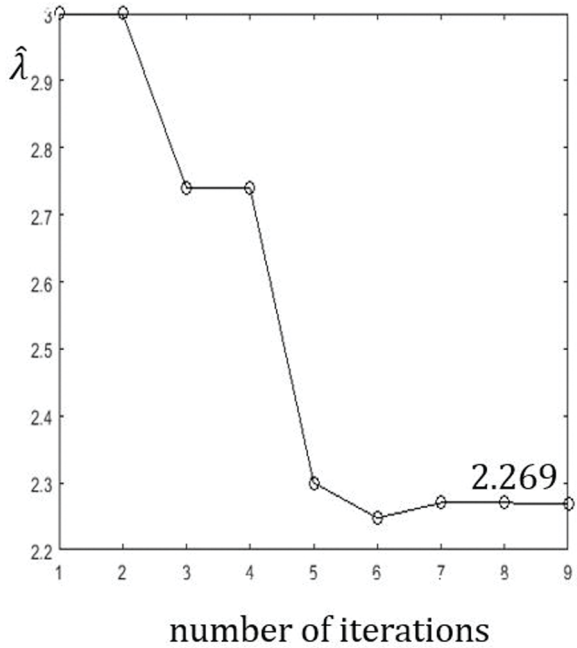


FIGURE 3.11b (Continued)

Example 3.4. We solve the MLE problem of Example 3.2 by Powell's method of conjugate directions.

Solution. With $n = 2$, computations start with the first stage of iterations at $\boldsymbol{x}_0 = (3,3)^T$ and $\boldsymbol{d}_1 = (1,\ 0)^T$ and $\boldsymbol{d}_2 = (0,\ 1)^T$. For this example problem, convergence is achieved at the end of one stage of iterations with two conjugate directions generated during the iterative process. Figure 3.11 shows the result obtained by using the data $\boldsymbol{z}$ which is same as the one earlier utilized by the HJ, NM and Rosenbrock methods.

■

3.3.3 Derivative-free Method with Trust Region Strategy

Trust region approach (Powell 1970, Sorensen 1982, Nocedal and Wright 2006) is like a dual to a line search method. While in line search, a direction is chosen to seek an optimum step size, trust region method searches for a direction with the step size fixed. For an unconstrained optimization problem, the method proposes a local quadratic model $q(\boldsymbol{x})$ at each iteration to the original objective function $f(\boldsymbol{x})$. The model is expected to be close to $f(\boldsymbol{x})$ in a selected region – a region of trust – around the current $\boldsymbol{x}_k$ of k^{th} iteration. The trust region is normally a ball with radius Δ_k and centred around $\boldsymbol{x}_k$, i.e. $\boldsymbol{x} \in \left\{\boldsymbol{y}, \left\|\boldsymbol{y} - \boldsymbol{x}_k\right\| \le \Delta_k\right\}$ where $\|.\|$ is the Euclidean norm. It may be natural to associate the technique with derivative-based methods when the Hessian of $f(\boldsymbol{x})$ is available or computable. The strategy is equally

adaptable within a derivative-free framework using quadratic surrogate models. For example, generation of these models by interpolation is widely discussed in Powell (1996) and Marazzi and Nocedal (2002) and Conn et al. (2009). In any case, with the Hessian of $f(\boldsymbol{x})$ assumed to be available at hand, the local quadratic model $q(\boldsymbol{x})$ at $\boldsymbol{x}_k$ is given by:

$$q(\boldsymbol{x}) = f(\boldsymbol{x}_k) + \boldsymbol{g}(\boldsymbol{x}_k)^T(\boldsymbol{x} - \boldsymbol{x}_k) + \tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_k)^T\boldsymbol{H}(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k) \tag{3.30}$$

Thus, the minimization problem of $f(\boldsymbol{x})$ reduces to a sequence of trust region subproblems of the form:

minimize $q(\boldsymbol{x})$

$$\text{s.t. } \|\boldsymbol{x} - \boldsymbol{x}_k\| \le \Delta_k \tag{3.31}$$

The solution of Equation (3.31) gives the next iterate $\boldsymbol{x}_{k+1}$ at which a new trust region problem is formulated. To solve Equation (3.31), one may adopt the Lagrange multiplier method and use the KKT condition:

$$[\boldsymbol{H}(\boldsymbol{x}_k) + \lambda \boldsymbol{I}_n](\boldsymbol{x} - \boldsymbol{x}_k) = -\boldsymbol{g}(\boldsymbol{x}_k) \tag{3.32}$$

$\lambda \ge 0$ is the Lagrange multiplier associated with the constraint in Equation (3.31). $\boldsymbol{H}(\boldsymbol{x}_k) + \lambda \boldsymbol{I}_n$ may be positive definite or semi-definite. $\boldsymbol{I}_n$ is the $n \times n$ identity matrix. Once each subproblem is solved, Δ_k is updated to fix the new trust region. The basics of the method are often related to the works of Levenberg (1944) and Marquardt (1963) where they proposed a method to iteratively solve nonlinear least squares error minimization problems with objective function $f(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{m} r_i^2(\boldsymbol{x})$ with $\boldsymbol{x} \in \mathbb{R}^n$ and each $r_i(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$. $r_i(\boldsymbol{x}), i = 1, 2, \ldots, m$ normally represent error residuals. The method involves calculation of a step size:

$$\boldsymbol{d}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k = -\left[\boldsymbol{J}(\boldsymbol{x}_k)^T \boldsymbol{J}(\boldsymbol{x}_k) + \mu_k \boldsymbol{I}_n\right]^{-1} \boldsymbol{J}(\boldsymbol{x}_k)^T \boldsymbol{r}(\boldsymbol{x}_k) \tag{3.33}$$

where $\boldsymbol{J}(\boldsymbol{x}_k) = \left[\dfrac{\partial r_i}{\partial x_j}\right]_{m \times n}$ is the Jacobian matrix evaluated at $\boldsymbol{x}_k$ and $r(\boldsymbol{x}_k) = \left(r_1(\boldsymbol{x}_k), r_2(\boldsymbol{x}_k), \ldots, r_m(\boldsymbol{x}_k)\right)^T$. μ_k is called a damping parameter devised to overcome the ill-conditioning of $\boldsymbol{J}(\boldsymbol{x}_k)$ and adjusted at each iteration. Notice that with $\mu_k = 0$, Equation (3.33) reduces to solving for $\boldsymbol{x}_{k+1}$ for each $r_i(\boldsymbol{x}_k)$ by Newton-Raphson method. For some $\Delta_k \in \mathbb{R}$, Equation (3.33) indicates that the vector $\boldsymbol{d}_k$ solves the problem:

$$\text{minimize: } \frac{1}{2}\left\|\boldsymbol{J}(\boldsymbol{x}_k)\boldsymbol{d} + r(\boldsymbol{x}_k)\right\|^2$$

$$\text{s.t. } \|\boldsymbol{d}\| \le \Delta_k \tag{3.34}$$

The corresponding quadratic model is $\frac{1}{2}\left\|\boldsymbol{r}\left(\boldsymbol{x}_k\right)\right\|^2+\boldsymbol{d}^T\boldsymbol{J}\left(\boldsymbol{x}_k\right)^T r\left(\boldsymbol{x}_k\right)+\frac{1}{2}\boldsymbol{d}^T\boldsymbol{J}\left(\boldsymbol{x}_k\right)^T \boldsymbol{J}\left(\boldsymbol{x}_k\right)\boldsymbol{d}$ which shows similarity to Equation (3.30). Taking cue from the works of Levenberg (1944) and Marquardt (1963), Powell (1970) proposed the first trust region algorithm for solving an unconstrained nonlinear optimization problem.

Update of the trust region step size Δ_k

If the local quadratic model $q(\boldsymbol{x})$ in Equation (3.30) yields a result that pertains to a reduced $f(\boldsymbol{x})$, the step size Δ_k is enlarged and the next trust region is constructed around $\boldsymbol{x}_{k+1}$. Otherwise, the region is contracted around $\boldsymbol{x}_{k+1}$ and the iterations are continued till convergence. The updating procedure is more precisely described with a parameter R_k defined as the ratio of the actual reduction in $f(\boldsymbol{x})$ to the predicted reduction in $q(\boldsymbol{x})$, i.e.:

$$R_k=\frac{f\left(x_k\right)-f\left(x_{k+1}\right)}{q\left(x_k\right)-q\left(x_{k+1}\right)} \tag{3.35}$$

While $R_k \cong 1$ indicates that the local quadratic approximation is quite reliable, $R_k<0$ shows no improvement in the iteration. With bounds $\varepsilon_1, \varepsilon_2 \in (0,1)$ specified on R_k, we expand the trust region radius Δ_k to, say, $2\Delta_k$ if $R_k > \varepsilon_1$. We reduce Δ_k to $\frac{\Delta_k}{4}$, if $R_k < \varepsilon_2$. For other values of R_k, Δ_k is unchanged. Typical values of ε_1 and ε_2 are 0.75 and 0.25, respectively. For other values of R_k, the radius Δ_k is unchanged. Iterations are continued with $\boldsymbol{x}_{k+1}=\boldsymbol{x}_k$ if the ratio is negative. Otherwise $\boldsymbol{x}_{k+1}$ is updated as $\boldsymbol{x}_k+\Delta_k$. Figure 3.12 shows a possible evolution of the trust regions within the design variable space of the unconstrained optimization problem.

It suffices to have an approximate solution to each trust region subproblem within its radius Δ_k. If the Cauchy's steepest descent method (Section 2.2.1, Chapter 2) is used, one obtains an approximate solution after setting $\boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{0}$ as:

$$\boldsymbol{d}_k^C=-\frac{\Delta_k \boldsymbol{g}(\boldsymbol{x}_k)}{\|\boldsymbol{g}(\boldsymbol{x}_k)\|} \tag{3.36}$$

Another alternative to obtain an improved solution is to use an exact $H(\boldsymbol{x})$ if available or an approximate $H(\boldsymbol{x})$ of quasi Newton method with, say, BFGS formula (Equation 2.85 in Chapter 2). Thus, the curvature of $q(\boldsymbol{x})$ is utilized in obtaining a better approximation $\boldsymbol{d}_k^B=-\boldsymbol{H}(\boldsymbol{x}_k)\boldsymbol{g}(\boldsymbol{x}_k)$. On the other hand, in arriving at $\boldsymbol{d}_k^C$, the quadratic term in $q(\boldsymbol{x})$ is avoided, and the solution so obtained may be a good approximation to the true one for small Δ_k. The Powell's dogleg method [1970] uses a direction made up of two line segments. The first one is along the steepest descent direction up to d_k^C and the second one along the line from d_k^C to d_k^B. The method minimizes $q(\boldsymbol{x})$ along this path which resembles a dog leg thus deriving its name. Refer to Nocedal and Wright (2006) for an analytical computation of the local minimizer for each trust region subproblem by the dogleg method without the necessity to perform a search along the chosen path.

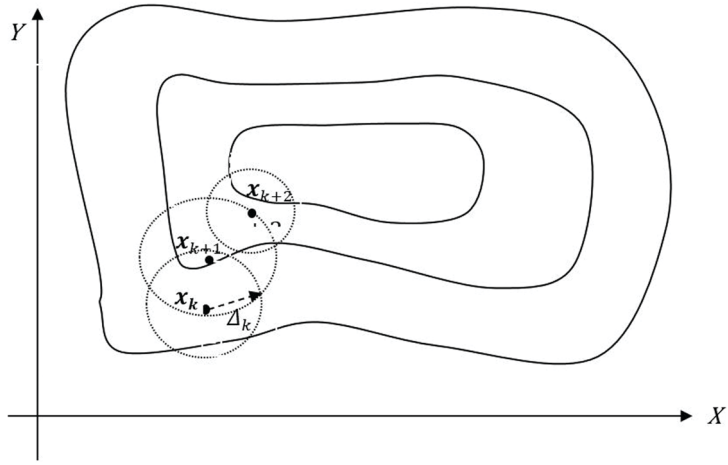


FIGURE 3.12 Trust region method in the two-dimensional case; evolution of trust regions along with new iterates $\boldsymbol{x}_i, i = k, k+1, k+2$.

If the original nonlinear problem is constrained, one may take recourse to one of the methods in Chapter 2 to transform the problem into an unconstrained one and then iteratively solve the latter using the local quadratic approximation – the main ingredient of the present method. The trust region methods has strong global convergence properties (More and Sorensen 1981, Sorensen 1982) with the sequence $\{\boldsymbol{x}_k\}$ converging to $\boldsymbol{x}^*$ so that $\boldsymbol{g}(\boldsymbol{x}^*)$ and $H(\boldsymbol{x}^*)$ are positive semi-definite. We refer to Toint (1988), Steihaug (1983), Conn et al. (2000) and Hager (2001) for examples on large-scale trust region subproblems.

Example 3.5. We solve the constrained optimization problem corresponding to the Rosen-Suzuki function defined in Equations (2.128) by the trust region method in conjunction with Nelder and Mead method.

Solution. The optimization problem in Equation (2.128) contains nonlinear constraints and is transformed into a sequence of unconstrained optimization problems by augmented Lagrangian method described in Section 2.4.3 of Chapter 2. Equation (2.129) gives the unconstrained and augmented objective function with μ_1 and μ_2 the Lagrange multipliers associated with the specified equality constraints and γ the corresponding multiplier for the inequality constraint. The equation is restated in the following.

$$\widehat{f}\left(\mu_1,\mu_2,\lambda,r_j,\boldsymbol{x}\right) = f(\boldsymbol{x}) + \sum_{i=1}^{2}\mu_i h_i(\boldsymbol{x}) + r_j\sum_{i=1}^{2}h_i^2(\boldsymbol{x}) + \gamma\max\left(g(x),-\frac{\gamma}{2r_j}\right) + r_j\left(\max\left(g(\boldsymbol{x}),-\frac{\lambda}{2r_j}\right)\right)^2 \tag{3.37}$$

where $f(\boldsymbol{x})$ is the Rosen-Suzuki function given by:

$$f(\boldsymbol{x}) = x_1^2 - 5x_1 + x_2^2 - 5x_2 + 2x_3^2 - 21x_3 + x_4^2 + 7x_4 + 50 \tag{3.38}$$

$h_i, i = 1,2$ and $g(\boldsymbol{x})$ are the equality and inequality constraints (see Equations 2.128b,c,d), respectively. Each unconstrained optimization of the non-quadratic $\hat{f}$ in Equation (3.37) corresponds to an increasing sequence of the penalty parameter $r_{j+1} = jr_j$. The computations are started with $r_1 = 1$ and the initial vector $\boldsymbol{x}_1 = (-1,1,-1,1)^T$. For each r_j, $\hat{f}$ is minimized by the trusted region method, i.e. by forming the quadratic function $q(\boldsymbol{x})$ at $\boldsymbol{x}_j$ corresponding to each r_j. The starting value for the trust region radius is $\Delta_0 = 2.0$. The minimization of $\hat{f}$ involves an iterative process. At each iteration k, the local quadratic model $q(\boldsymbol{x}_k)$ is minimized by Nelder and Mead method and trust region updated according to Equation (3.35). Hessian $\boldsymbol{H}(\boldsymbol{x}_k)$ and gradient $\boldsymbol{g}(\boldsymbol{x}_k)$ of $\hat{f}$ are updated at each iteration before forming $q(\boldsymbol{x}_k)$. Both $\boldsymbol{g}(.)$ and $\boldsymbol{H}(.)$ are computed at each iteration by a finite difference scheme (Section 2.4.4, Chapter 2). Figures 3.13a–b show the result obtained by trust region method along with Nelder and Mead method.

One may directly apply Newton's method to get the local optimizer to $q(\boldsymbol{x})$ and obtain the new iterate $\boldsymbol{x}_{k+1}$ as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - [\boldsymbol{H}(\boldsymbol{x}_k)]^{-1}\boldsymbol{g}(\boldsymbol{x}_k) \tag{3.39}$$

This as an approximate solution to Equation (3.30) and violation of the box constraints $\|\boldsymbol{x} - \boldsymbol{x}_k\| \leq \Delta_k$ may be checked before updating the trust region radius as per the criterion in Equation (3.35). Here, results are also obtained (Figures 3.14a–b) by Powell's method of conjugate directions by solving the non-quadratic objective function in Equation (3.37) corresponding to each r_j.

■

3.4 METAHEURISTICS – EVOLUTIONARY METHODS

As pointed out in the introduction to this chapter, derivative-free methods might be the only option left in the absence of a generic directional information (equivalent to the Gateaux derivative, a generalization of a directional derivative in the search for a local extremum of a sufficiently smooth cost functional). In this regard, evolutionary stochastic search techniques (e.g. the genetic algorithm [Goldberg 1989, Koza 1992]) have proved very effective compared to the gradient-based deterministic counterparts or even the derivative-free deterministic search methods such as the HJ and NM methods. This is in fact true (Fletcher 1987, Chong and Zak 2013) even for cases involving sufficiently smooth, yet multimodal, objective/cost functionals,

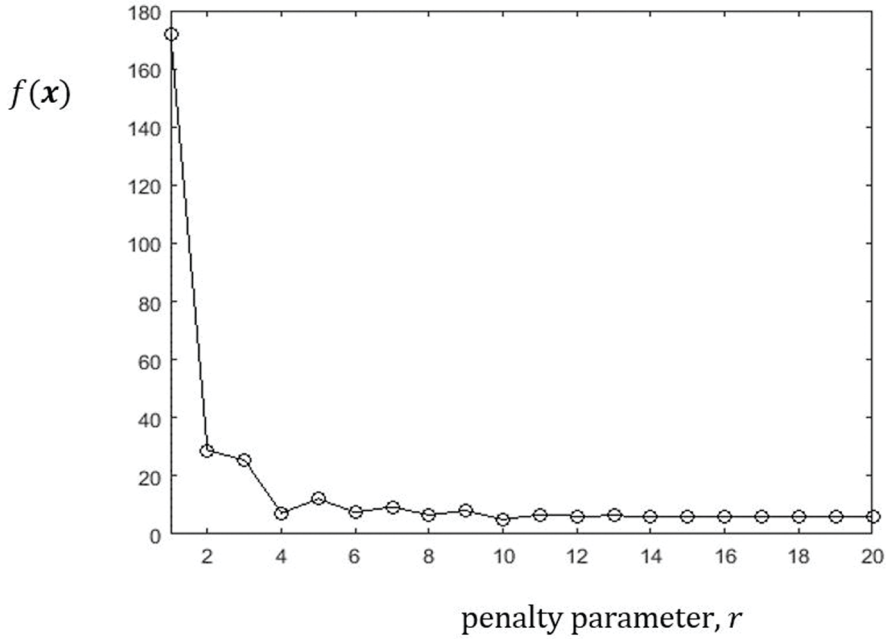


FIGURE 3.13a Solution to constrained optimization problem in Example 3.5 by trust region method combined with Nelder and Mead method, $r_0 = 1$; evolution of $f(\boldsymbol{x})$ with respect to r (finally attaining a minimum value of 6.19).

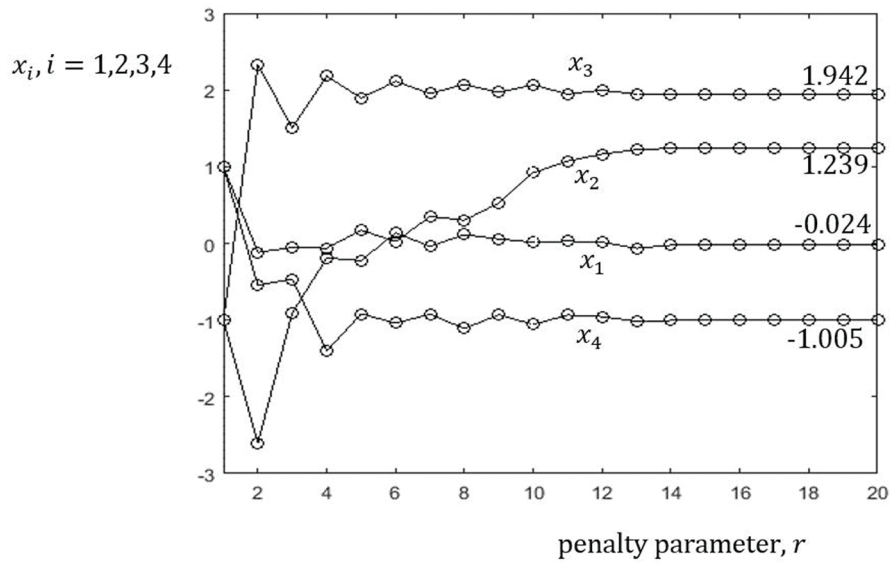


FIGURE 3.13b Solution to the constrained optimization problem in Example 3.5 by trust region method combined with Nelder and Mead method, $r_0 = 1$; evolutions of design variables $x_i, i = 1,2,3,4$ with respect to r.

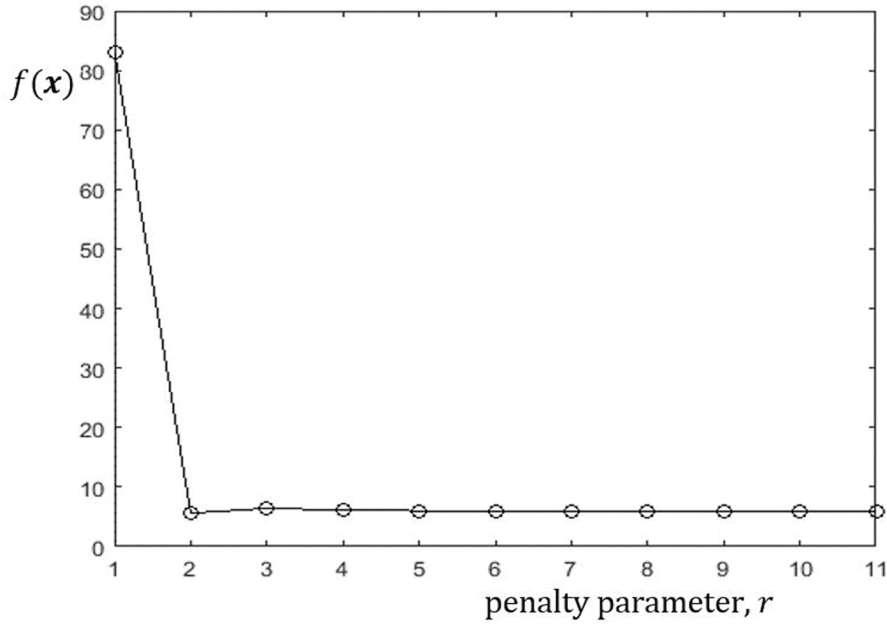


FIGURE 3.14a Solution to constrained optimization problem in Example 3.5 by trust region method combined with Powell's method of conjugate directions, evolution of $f(x)$ with respect to r (finally attaining a minimum value of 6.009).

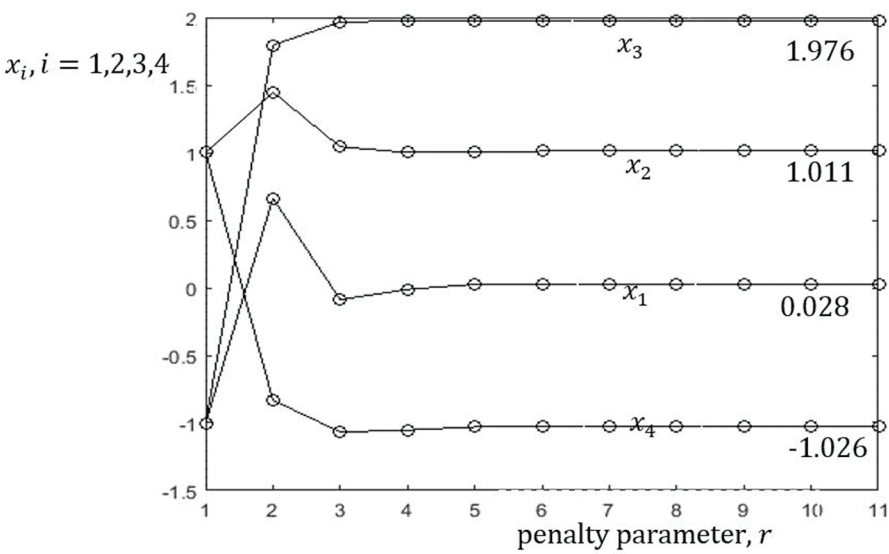


FIGURE 3.14b Solution to the constrained optimization problem in Example 3.5 by trust region method combined with Powell's method of conjugate directions; evolutions of $x_i, i = 1,2,3,4$ with respect to r.

wherein the use of directional derivatives may be inadequate in obtaining the global optimum. The suitability of a meta-heuristic method in attaining the global optimum within a finite time may only be numerically demonstrated in a problem-specific manner, even though it typically offers no guarantee of reaching the global solution. Notable schemes in this genre are the genetic algorithm (Holland 1975; Goldberg 1989, Daniel 2006), particle swarm optimization (Kennedy and Eberhart 1995, Shi and Eberhart 1998, Kennedy 2006), simulated annealing (Kirkpatrick et al. 1983, Van Laarhoven and Aarts 1987), differential evolution (Storn and Price 1997, Price et al. 2005), Tabu search (Glover and Laguna.1997) ant colony optimization (Dorigo et al. 1996, 2011, Slowik and Kwasnicka 2018) and the covariance matrix adaptation evolution strategy (Hansen and Ostermeier 1996, Hansen 2007). Some of these techniques are described in the sections that follow.

3.4.1 Genetic Algorithm (GA)

GA is an evolutionary global optimization scheme generally meant to handle unconstrained optimization problems (Michalewicz and Schoenauer 1996, Deb 1997). It metaphorically 'mimics the genetic evolution of a species and follows the biological processes' with the ability to get adapted to the 'environment' in the consecutive generations, where a notion of competitiveness in this 'environment' is provided by the objective functional. For the words quoted above, no mathematically rigorous meaning should be attributed. The adaptation process is mainly accomplished in the form of genetic inheritance from parents to offspring and through the so-called survival of the fittest. In this method, the initial population is created randomly. Each element (candidate) in the population is called a chromosome,[§] which is constructed using the design variables. A candidate is characterized by a set of parameters (design variables) called 'genes'. The genes joined into a string forms a chromosome.

For the update, the objective function assigns a measure of competitiveness or goodness to the chromosomes in terms of their fitness. Like most evolutionary schemes, GA also relies upon 'exploration-exploitation trade-off' which is implemented using three operators, 'crossover', 'mutation' and 'selection'. While 'crossover' and 'mutation' impart variations in the chromosomes, the selection operator is used to choose chromosomes for the subsequent population. Chromosomes with higher fitness values have higher chances of being selected. For crossover, pairs of parents, which participate in mating to produce offspring, are chosen based on some probability.

To impart additional layers of variation in the population, the mutation operator is applied to a randomly chosen chromosome to alter one or more of its genes. This is

§ Chromosome
Chromosomes are strings of genes in living organisms. By Darwin's theory about evolution, all living organisms consist of cells and each cell consists of the same set of chromosomes. The set serves as a model for the whole organism. Genetic algorithm adopts this model and in this context, chromosome stands for a set of design variables. It is a possible solution (candidate) to the optimization problem. A set of solutions represented by chromosomes is called a population

again based on a specified probability. Before the selection operation is carried out, one may also apply one penultimate operation known as 'elitism' to the candidates at each iteration. This is meant to retain the best candidates (some percentage of the total population) unchanged and carry over to the next iteration. The operation is expected to keep the good candidates that may be lost due to the cross-over and mutation operations.

Each candidate in the initial population is represented by a string. The string has different segments replicating the genes in a chromosome. The number of genes (segments) in a string depends on the number of design variables needing representation. If we have $\boldsymbol{x} = \left(x_1, x_2\right)^T$ in a two-dimensional problem, a string consists of two genes – the first representing x_1 and the second x_2. Each gene is expressed in terms of a fixed-length binary sequence of bits. The number of bits in the i^{th} gene is determined by:

$$l_j = \log_2 \left(\frac{x_{j,u} - x_{j,l}}{s_j} \right), \; j = 1, 2, ..., n \tag{3.40}$$

where n is the number of design variables. $x_{j,u}$ and $x_{j,l}$ are the upper and lower bounds of the design variable x_j. s_j is the desired precision for the variable. The binary bits are generated by a random number generator. If the generated number is less than 0.5, we assign zero to the bit and if greater than 0.5, it is assigned unity. Table 3.3 explains the basic steps – crossover and mutation – of GA.

The selection operator that completes the basic structure of the GA is used to select the survived chromosomes, again based on some probability, for the next iteration. A stopping criterion is based on the convergence of the objective function within a certain tolerance vis-à-vis the previous value. Otherwise, the algorithm may also be stopped after a specified maximum number of iterations. The steps defining the stochastic search scheme of the GA are enumerated in Table 3.4.

Example 3.6. We find the solution to the Rosenbrock function $f(\boldsymbol{x}) = 100 \left(x_1^2 - x_2\right)^2 + \left(x_1 - 1\right)^2$ by GA.

Solution. The optimization problem is solved by taking the population size $N_p = 200$. The size is chosen after repeated trials with other values of $N_p < 200$ exhibited less consistency in realizing the optimum $\boldsymbol{x}^* = (1,1)^T$. The crossover and mutation probabilities are 0.5 and 0.01 respectively. Figure 3.15 shows the convergence of the objective function and the two variables x_1 and x_2.

TABLE 3.3
Crossover and Mutation Operations in GA Scheme

Consider a 2-dimensional optimization problem, i.e., the number of design variables or parameters $n = 2$. Initialize N_p, the population size.

Suppose $\boldsymbol{P}_i = (x_1^{[i]}, x_2^{[i]})$ and $\boldsymbol{P}_j = (x_1^{[j]}, x_2^{[j]})$, $i, j \in (1, N_p)$ are two parent candidates (chromosomes) of the population and they are represented in binary form (with each parameter by, say, an 8-bit string).

i^{th} chromozome - $\boldsymbol{P_i}$: 1 0 0 0 1 0 0 1 0 1 1 1 1 1 1 0 ($x_1^{[j]}$, $x_2^{[j]}$)

j^{th} chromozome - $\boldsymbol{P_j}$: 0 1 0 0 0 1 0 0 1 1 1 0 0 1 1 1 ($x_1^{[j]}$, $x_2^{[j]}$)

Each bit represents a gene in the cromozome.

a) single-point crossover operation:

$\boldsymbol{P_i}$: 1 0 0 0 1 | 0 0 1 0 1 1 1 1 1 1 0 parent 1

$\boldsymbol{P_j}$: 0 1 0 0 0 | 1 0 0 1 1 1 0 0 1 1 1 parent 2

set of genes selected for crossover

$\boldsymbol{C_i}$: 1 0 0 0 1 | 1 0 0 1 1 1 0 0 1 1 1 offspring 1

$\boldsymbol{C_j}$: 0 1 0 0 0 | 0 0 1 0 1 1 1 1 1 1 0 offspring 2

b) multi-point crossover operation:

$\boldsymbol{P_i}$: 1 0 0 0 1 | 0 0 1 0 1 | 1 1 1 | 1 1 0 parent 1

$\boldsymbol{P_j}$: 0 1 0 0 0 | 1 0 0 1 1 | 1 0 0 | 1 1 1 parent 2

sets of genes selected for crossover

$\boldsymbol{C_i}$: 1 0 0 0 1 | 1 0 0 1 1 | 1 1 1 | 1 1 1 offspring 1

$\boldsymbol{C_j}$: 0 1 0 0 0 | 0 0 1 0 1 | 1 0 0 | 1 1 0 offspring 2

c) mutation operation (assuming single point crossover)

$\boldsymbol{C_i}$: 1 0 0 0 1 1 0 0 1 1 1 | 1 1 1 1 1 offspring 1

$\boldsymbol{C_j}$: 0 1 0 0 0 0 0 1 0 1 1 | 0 0 1 1 0 offspring 2

gene selected randomly

$\boldsymbol{C_i}$: 1 0 0 0 1 1 0 0 1 1 1 | 0 1 1 1 1 offspring 1

$\boldsymbol{C_j}$: 0 1 0 0 0 0 0 1 0 1 1 | 1 0 1 1 0 offspring 2

TABLE 3.4
Main Steps in the Stochastic Search Algorithm of GA

Step 1. Generate the initial population by randomly selecting the genes of each candidate (chromosome). Each chromosome stands for a candidate (realized) solution of the problem.
Step 2. Find the fitness value (based on the objective function) of each chromosome of the initial population.
Step 3. Divide the population into sets of parents by a random combination among the members of the population.
Step 4. Crossover operation: Get the offspring by exchanging the corresponding genes in the parent members by single point or multi-point crossover (see Table 3.3). The crossover points are randomly selected.
Step 5. Mutation operation: Randomly change the value of a selected gene in some of the offspring. The member and the gene to be changed are randomly selected.
Step 6. Perform elitism operation with a fixed percentage of the population.
Step 7. Select the new generation (i.e. retain or reject the parent chromosomes) based on the fitness value of the offspring (new chromosomes).
Step 8. Go to step 3 and iterate till the stopping criterion is satisfied.

■

Other features of GA

For relatively better performance, particularly in engineering applications, use of real-valued (continuous) representation of the design variables is suggested (Haupt and Haupt 2004) instead of taking a recourse to encoding in the binary format. They may be normalized as $\overline{x}_j = \left(x_j - x_{j,l}\right)/\left(x_{j,u} - x_{j,l}\right),\ j = 1,2,..,n.$ so that all variables are in the interval [0,1]. For finding the fitness value of a member, the variable is unnormalized by having $x_j = \overline{x}_j\left(x_{j,u} - x_{j,l}\right) + x_{j,l}$. The initial population is randomly generated according as:

$$x_j^{[i]} = x_{j,l} + u\left(x_{j,u} - x_{j,l}\right) \tag{3.41}$$

where $u \sim U(0,1)$ is a uniformly distributed random number in [0,1]. The crossover operation may be accomplished by a simple swapping of variable values between a pair of chromosomes at random points. The other way is by blending two (normalized) parent members so as to obtain the new offspring member (Haupt 1995). Suppose that we have the parent chromosomes $\boldsymbol{P}_i = \left(x_1^{[i]}, x_2^{[i]}, \ldots, x_q^{[i]}, \ldots, x_n^{[i]}\right)$ and $\boldsymbol{P}_j = \left(x_1^{[j]}, x_2^{[j]}, \ldots, x_q^{[j]}, \ldots, x_n^{[j]}\right)$. We randomly select the q^{th} variable for crossover and reproduce two new variables as:

$$x_{q,new}^{[i]} = x_q^{[i]} + \beta\left(x_q^{[j]} - x_q^{[i]}\right)$$

$$x_{q,new}^{[j]} = x_q^{[j]} - \beta\left(x_q^{[j]} - x_q^{[i]}\right) \tag{3.42a,b}$$

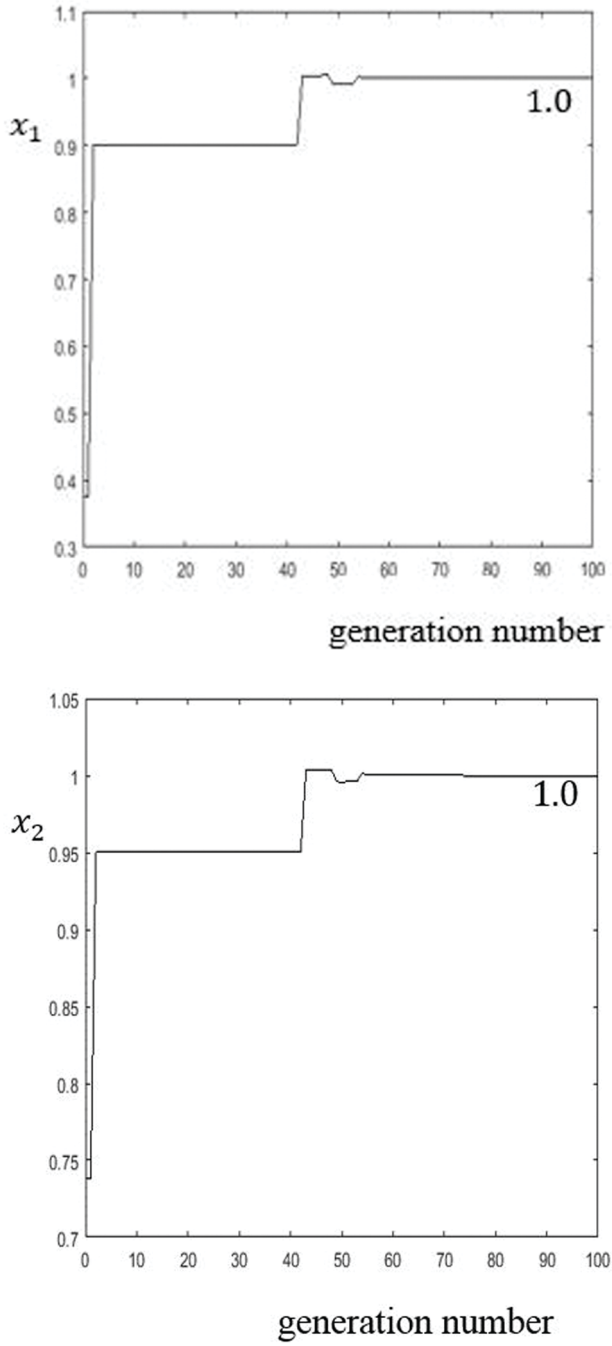


FIGURE 3.15a–b GA solution to Rosenbrock function $f(x) = 100\left(x_1^2 - x_2\right)^2 + \left(x_1 - 1\right)^2$; crossover probability = 0.2, mutation probability = 0.2: (a–b) evolution of x_1 and x_2 with iterations.

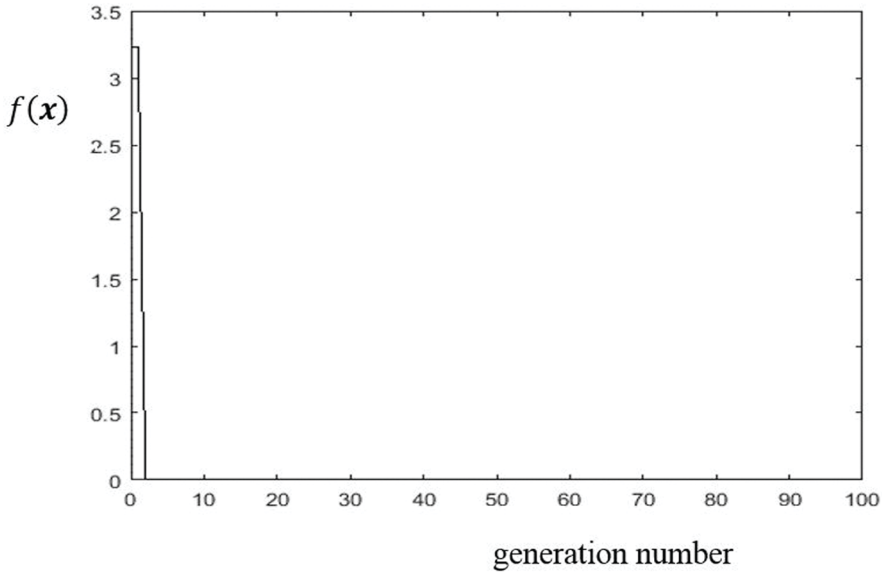


FIGURE 3.15c GA solution to Rosenbrock function $f(\boldsymbol{x}) = 100\left(x_1^2 - x_2\right)^2 + \left(x_1 - 1\right)^2$; crossover probability = 0.2, mutation probability = 0.2, evolution of the objective function with iterations (finally attaining a minimum value of 6.063E-7).

$\beta \in [0,1]$ is the blending parameter. Mutation operation is performed by randomly selecting small fraction of chromosomes in the population. In each of these candidates, a variable x_q is randomly selected and replaced by a new value according to Equation (3.41). This operation is executed over the selected chromosomes only when a mutation probability is satisfied. That is, with a mutation probability p_m, a random number $r \sim U(0,1)$ is selected at each iteration and the mutation operation is carried out only if $r \leq p_m$. The mutation probability is initially taken as unity and is gradually decreased as iterations progress. This helps in a wider exploration of the design space in the initial stages thus escaping from any local optima. Similarly, we retain the best parent members without mutation to have the elitism operation.

Constrained optimization problems can be handled by GA in conjunction with methods like augmented Lagrangian (Section 2.4.3, Chapter 2). The latter as described in Chapter 2 converts the constrained problem into a sequence of unconstrained ones using a penalty parameter.

Example 3.7. Here we use the GA combined with the augmented Lagrangian method to solve the constrained optimization problem corresponding to the Rosen-Suzuki function defined in Equations (2.128).

Solution: Equation (2.129) gives the unconstrained and augmented objective function with μ_1 and μ_2 being the Lagrange multipliers associated with the specified equality constraints and λ the corresponding multiplier for the inequality constraint. Each

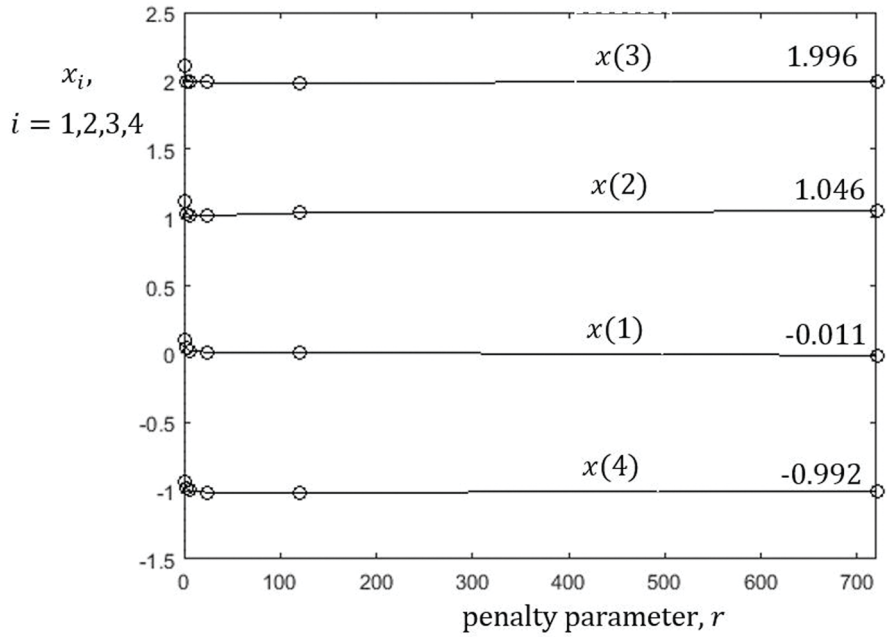


FIGURE 3.16a Solution to constrained optimization of Rosen-Suzuki function by GA plus augmented Lagrangian method, $N_p = 100$, mutation rate = 0.005; convergence of the four design variables $x_i, i = 1,2,3$ and 4.

unconstrained optimization corresponding to the decreasing sequence of the penalty parameter r_k is solved by the GA. The results obtained by a combination of the two methods are given in Figures 3.16a–b.

■

There are variants of the GA (Srinivas and Patnaik 1994, Smith and Fogarty 1997) involving, for example, self-adaptation of the key parameters – cross-over and mutation rates. GA has been applied to a diverse range of optimization problems (Haupt 1995, Back 1996). These include structural optimization (Goldberg and Samtani 1986, Doorly et al. 1996, Eby et al. 1999, Guerlement et al. 2001, Deb and Gulati 2000, Hultman 2010), control systems optimization (Krishnakumar and Goldberg 1992) and machine learning (Michalewicz 1996). As a global optimization tool, GA is particularly well-suited for parallelism (Alba and Tomassini 2002) due to its ability of simultaneously exploring the search space in multiple directions. An example for optimum choice of resonant frequencies of a structural system using GA is illustrated below.

Example 3.8. We consider the dynamics of a straight circular shaft supported on springs and optimize the shaft geometry so as to have the first two natural frequencies

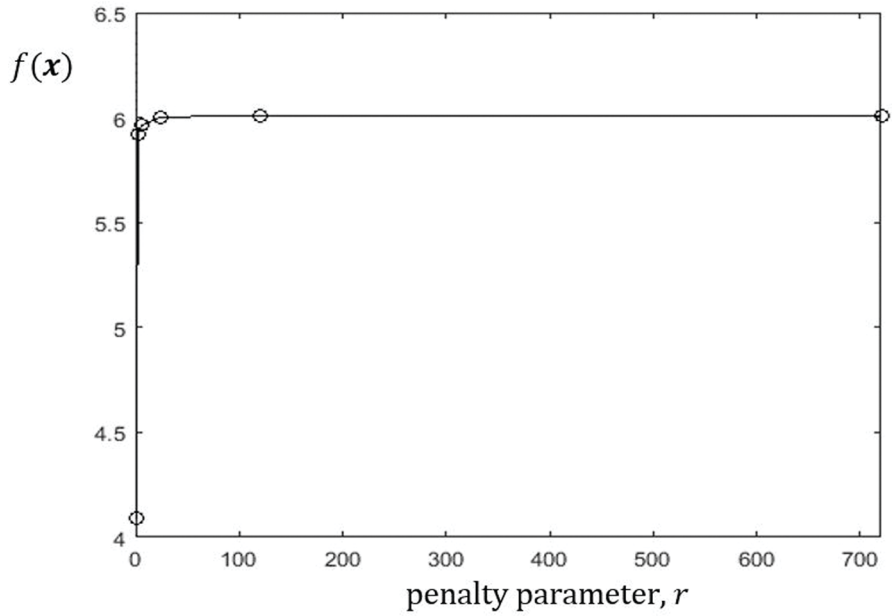


FIGURE 3.16b Solution to the constrained optimization of Rosen-Suzuki function by GA plus augmented Lagrangian method; N_p = 100 mutation rate = 0.005; convergence of the objective function with respect to the penalty parameter r (finally attaining a minimum value of 6.008).

separated by a desired frequency range. The objective is to avoid any resonance (Appendix 3) in the specified frequency range and thus avoid possible high amplitudes of vibration. GA is used to achieve the objective.

Solution. The circular shaft is shown in Figure 3.17a. The supporting springs have the stiffness of 4.378 N/m in both the transverse directions Y and Z. The shaft, whose length is $l = 0.3\ m$, has a disk at the left end with mass equal to $1.4\ kg$ The Young's modulus of elasticity E of the shaft material is $2.075 \times \frac{10^{11} N}{sq}.\ m$ and the mass density $\rho = 7806 \frac{kg}{m^3}$. The FEM is used to discretize (Figures 3.17b and 3.17c) the shaft and arrive at the dynamic equations of motion. We refer to Section 1.5.1 for a description of the FEM.

The FE model, shown in Figure 3.17b, consists of one-dimensional beam elements with 4 dofs/node – two translational and two rotational. A typical beam element is shown in Figure 3.17c. Since each beam element is two-noded, the element mass and stiffness matrices $\boldsymbol{M}^e$ and $\boldsymbol{K}^e$ [Bathe 1998] are of size 8×8. The total number

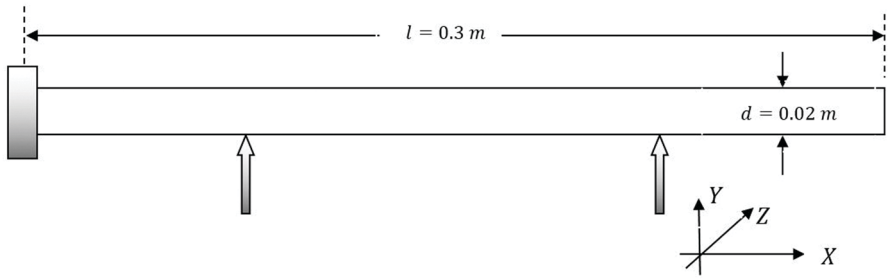


FIGURE 3.17a Spring-supported circular shaft.

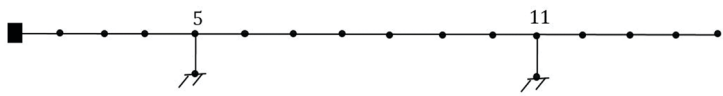


FIGURE 3.17b Spring-supported shaft and the FE model with beam elements.

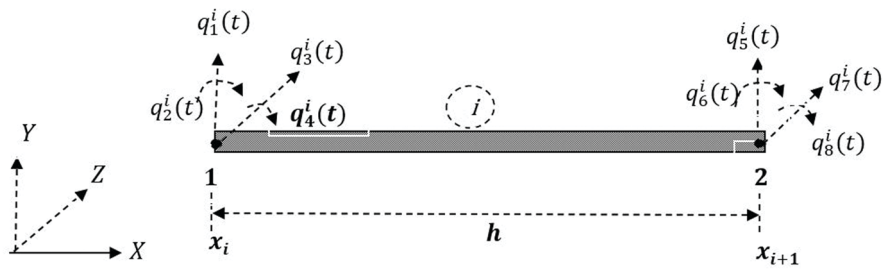


FIGURE 3.17c Spring-supported shaft and a typical beam element (i^{th}) with 4 dofs per node: $q_1^i(t), q_3^i(t), q_5^i(t), q_7^i(t)$ – translational *dof* and $q_2^i(t), q_4^i(t), q_6^i(t), q_8^i(t)$ – rotational dof.

of beam elements in the FE model is $N_e = 15$ and the number of active *dofs* is $N = 4\times$ number of nodes on the shaft = 64. The disk at the left end of the shaft is modelled by a lumped mass element. The design variables are the shaft diameters $\{x_j, j = 1,2,\ldots,15\}$ denoted by the vector x. The lower and upper bounds for the diameters are uniformly taken to be 0.004 m and 0.05 m, respectively.

After the final assembly of element mass and stiffness matrices, we obtain the equations of motion, in the form of ODEs, may be written in the following matrix-vector form:

$$\boldsymbol{M}\ddot{\boldsymbol{y}}(t) + \boldsymbol{K}\boldsymbol{y}(t) = \boldsymbol{\mathcal{P}}(t) \tag{3.43a}$$

$\boldsymbol{y}(t) = \left(y_1(t), y_2(t), \ldots, y_N(t)\right)^T$ is the vector of nodal displacements (translational and rotational). $\boldsymbol{M}$ and $\boldsymbol{K}$ are the $N \times N$ assembled mass and stiffness matrices (resp.) and $\boldsymbol{\mathcal{P}}(t) \in \mathbb{R}^N$ is the vector of nodal forces. Equation (3.43a) corresponds to an undamped system. As energy dissipation is invariably present in practice, we assume Rayleigh damping, a form of viscous damping expressible as a weighted linear combination of the mass and stiffness matrices (Clough and Penzien 1982). Accordingly, the damping matrix $\boldsymbol{C}$ is constructed in the form $\boldsymbol{C} = \alpha\boldsymbol{M} + \beta\boldsymbol{K}$ with $\alpha, \beta \in \mathbb{R}$ and we write the damped equations of motion as:

$$\boldsymbol{M}\ddot{\boldsymbol{y}}(t) + \boldsymbol{C}\dot{\boldsymbol{y}}(t) + \boldsymbol{K}\boldsymbol{y}(t) = \boldsymbol{\mathcal{P}}(t) \tag{3.43b}$$

The natural frequencies (Appendix 3) of the shaft may be obtained by studying the frequency response (also described in Appendix 3) due to a harmonic excitation of the form $\boldsymbol{\mathcal{P}}(t) = \boldsymbol{A} \sin \lambda$.

$\boldsymbol{A}$ is the vector of excitation amplitudes. λ is the excitation frequency in rad/s. When the shaft is excited only at a mass point (node 1 in the FE model – Figure 3.17b) in the Y and Z directions, $\boldsymbol{\mathcal{P}}(t) = \left(A_1 \sin \lambda t,\ A_2 \sin \lambda t, 0, 0, \ldots\right)^T$. With the initial guess of a uniform diameter of $0.02\ m$ for the elements, the frequency response at the support points of the shaft (prior to the start of iterations) is shown in Figure 3.18. The shaft being of circular cross-section, the response in two transverse (Y and Z) directions is identical at each support point.

In obtaining the frequency response in Figure 3.18, λ is varied in the range $0 - 10000$ rad/s, keeping the excitation amplitudes A_1 and A_2 fixed at unity. From the figure, the first natural frequency ω_1 of the shaft is identified to be 1250 rad/s. The objective of optimization is presently to modify the shaft geometry, i.e. the shaft diameters $x_j, j = 1, 2, \ldots, 15$ so as to push the natural frequencies to outside the range 1000 – 2000 rad/s, which is the range over which loading frequencies could vary. The aim is then to minimize the error as in the following objective function:

$$f(\boldsymbol{x}) = \left(1000 - \omega_1(\boldsymbol{x})\right)^2 + \left(2000 - \omega_2(\boldsymbol{x})\right)^2 \tag{3.44}$$

While the objective function does not directly depend on the shaft diameters, the first two natural frequencies ω_1 and ω_2 are functions of x. GA is used with a population size of $N_p = 20$. The mutation rate is 0.01 and the mutation probability p_m is unity initially as it is progressively reduced during the iterative process based on $p_{m,k+1} = cp_{m,k}$ with $c \in (0,1)$. Figures 3.19a–b show the results.

Since GA is a stochastic search algorithm, the results shown in Figure 3.19 are sample-averaged quantities obtained at each iteration by averaging over a population of $N_p = 20$. Before the start of iterations, search for the optimum begins with each of the design variables (here shaft diameters $x_j, j = 1, 2, \ldots, n = 15$) selected according to a uniform probability distribution (Equation 3.41). Thus, each x_j is a random

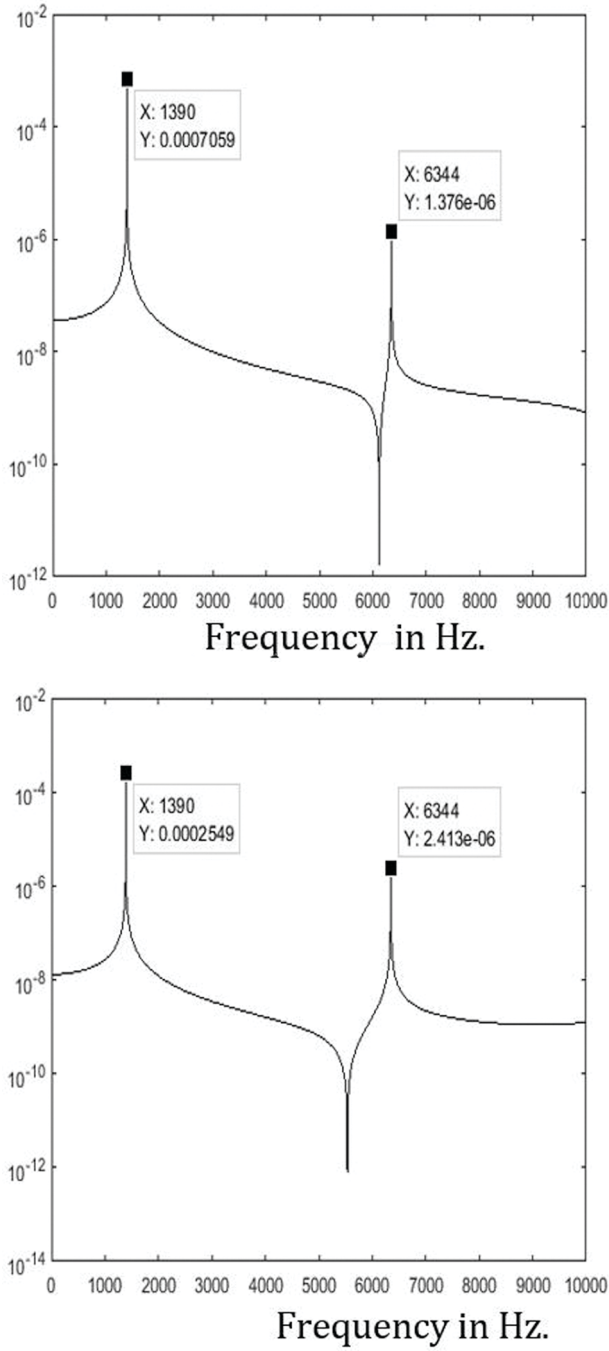


FIGURE 3.18a–b Frequency response before start of iteration in Example 3.8 at the support points in *Y*- and *Z*-directions; excitation amplitudes $A_1, A_2 = 1.0$ N at the disk node: (a) response at node 5 – in *Y*-direction and (b) response at node 11 – in *Y*-direction (see FE model in Figure 3.17b).

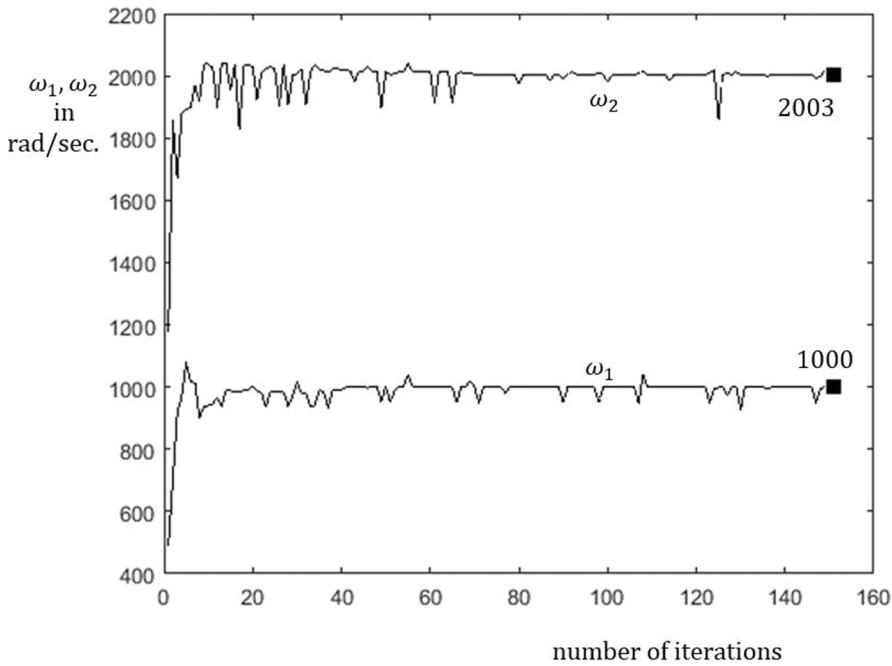


FIGURE 3.19a Optimum shaft geometry by GA to avoid resonance in a specified frequency range; $N_p = 20$, $p_{m,k+1} = 0.99\, p_{m,k}$; evolutions of the first two natural frequencies ω_1 and ω_2.

variable and its distribution is altered at each iteration. For clarity, the evolutions of x_3 and x_7 are shown in Figures 3.20a and 3.20b, respectively, show in the first few iterations. Following this, Figures 3.20c and 3.20d, respectively, show the evolutions of these two diameters over the entire range of iteration. As observed, sample variances of the design variables reduce with iterations, finally converging to nearly deterministic values (Figures 3.20c and 3.20d). This is desirable since the original optimization problem is posed deterministically. Figure 3.21 shows the convergence of all the design variables (sample averaged over the population size $N_p = 20$) with iterations.

Note that the solution obtained may still be a local solution instead of global even though the present solution is sufficiently accurate. Fine tuning of the algorithmic parameters such as the population size, mutation rate and probability may realize a better solution. Figure 3.22 shows the frequency response of the mass located at the left end of the shaft, obtained with the final (optimized) set of shaft diameters at the end of iterations. The response corresponds to a transverse direction at the mass point. The result shows that the desired objective to avoid resonance in the frequency range 1000–2000 rad/s is indeed realized. The optimum shaft geometry is shown in Figure 3.23.

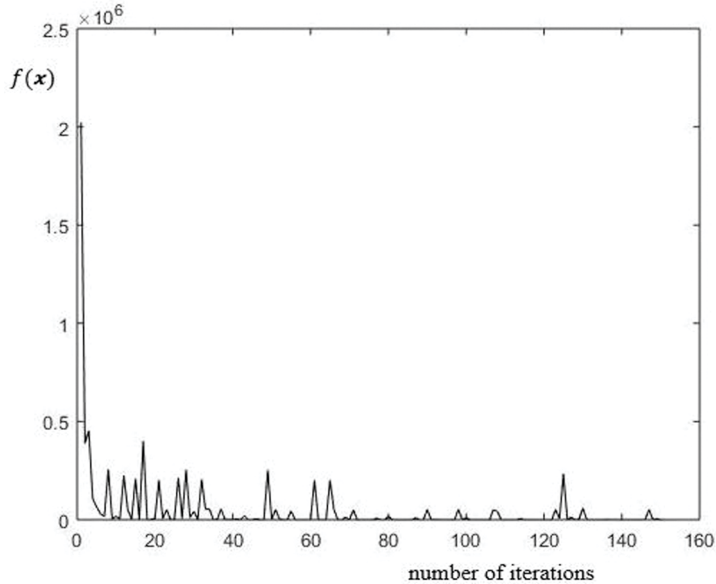


FIGURE 3.19b Optimum shaft geometry by GA to avoid resonance in a specified frequency range; $N_p = 20$, $p_{m,k+1} = 0.99\,p_{m,k}$; evolution of the objective function (in Equation 3.44) with iterations (finally attaining a minimum value of 9.0).

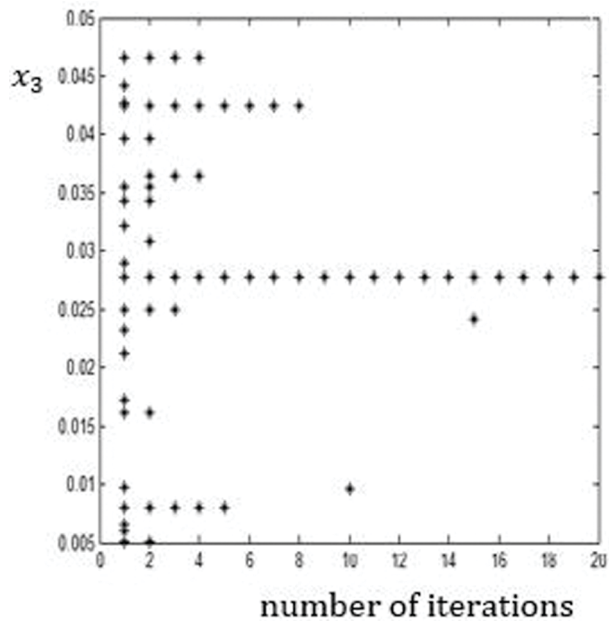
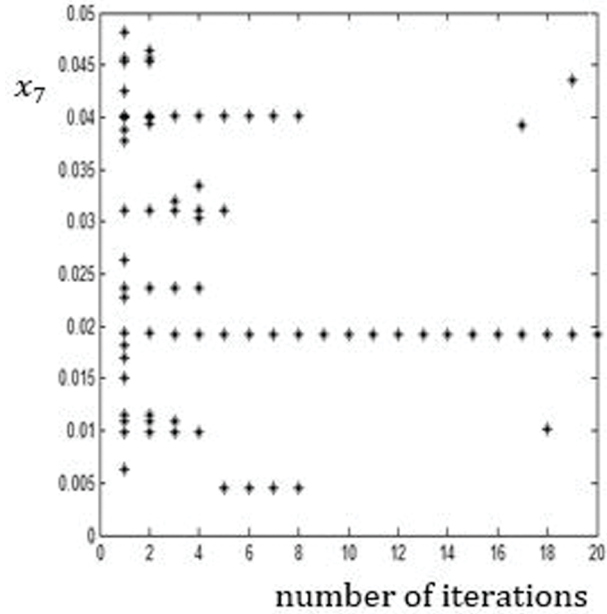


FIGURE 3.20a Optimum shaft geometry by GA to avoid resonance in a specified frequency range; $N_p = 20$, $p_{m,k+1} = 0.99\,p_{m,k}$: (a) evolution of x_3 over the first 20 iterations; (b) evolution of x_7 over the first 20 iterations; (c) evolution of x_3 over all iterations; (d) evolution of x_7 over all iterations.



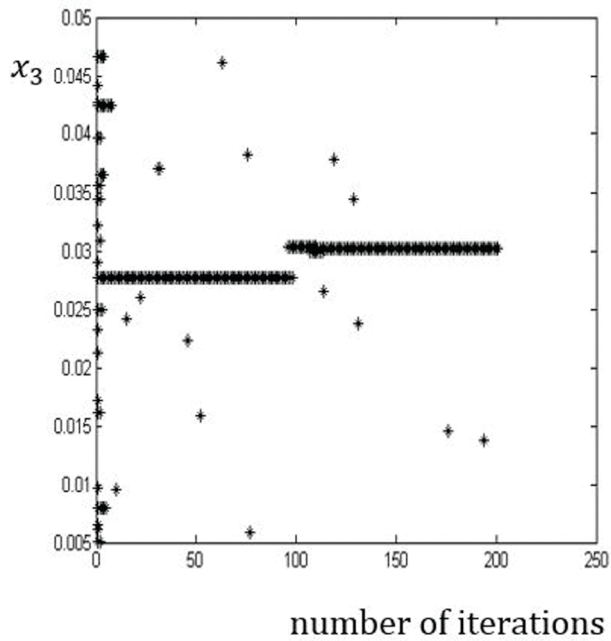


FIGURE 3.20b–c (Continued)

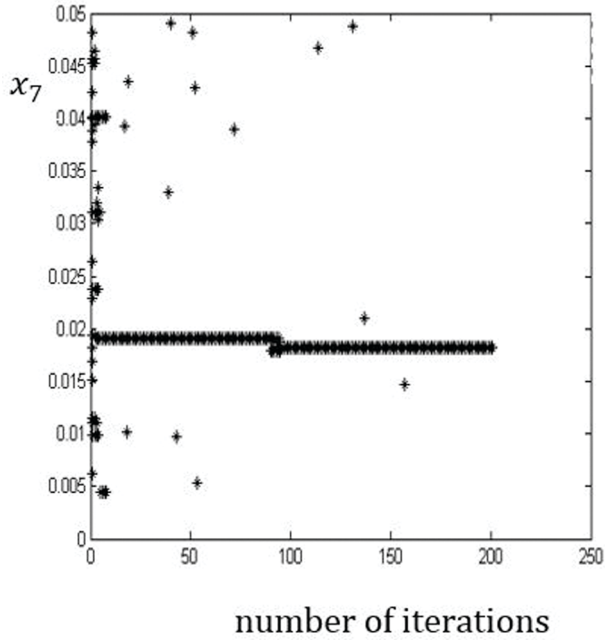


FIGURE 3.20d (Continued)

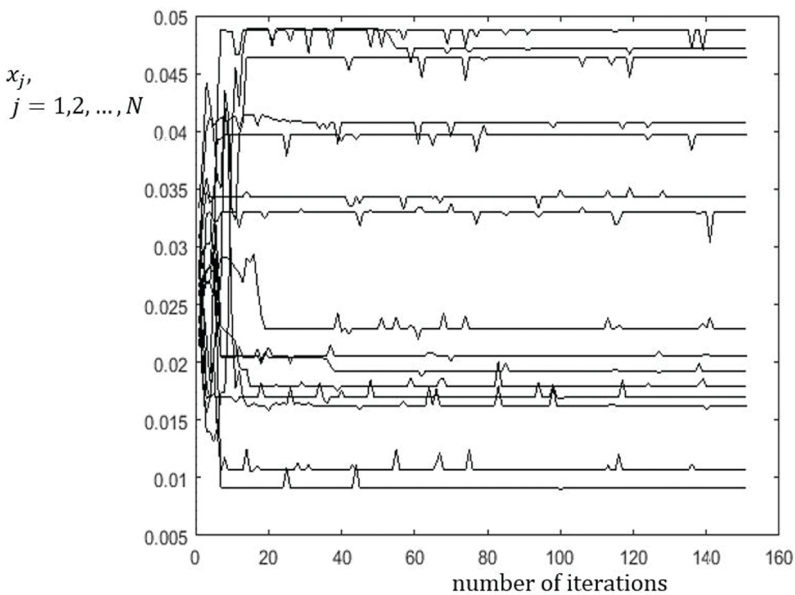


FIGURE 3.21 Optimum solution by GA for a simply supported shaft to avoid resonance in a specified frequency range; $N = 15$, $N_p = 20$, $p_{m,k+1} = 0.99 p_{m,k}$; evolutions of all design variables $x_j, j = 1,2,\ldots,N$ (sample-averaged) with iterations.

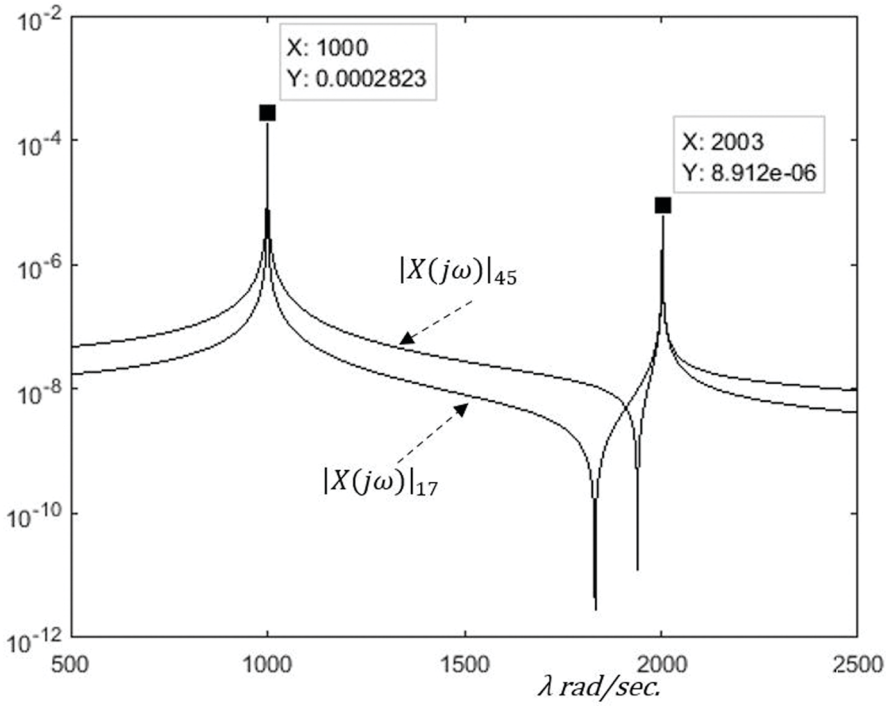


FIGURE 3.22 Optimum solution by GA for a simply supported shaft to avoid resonance in a specified frequency range; $N = 15$, $N_p = 20$, $p_{m,k+1} = 0.99 p_{m,k}$; frequency response of the shaft with the final set of diameters $x_j, j = 1,2,\ldots,N$ obtained by GA; $\left|X(j\omega)\right|_{17}$ – response at 5th node in *Y*-direction and $\left|X(j\omega)\right|_{45}$ – response at 11th node in *Y*-direction.

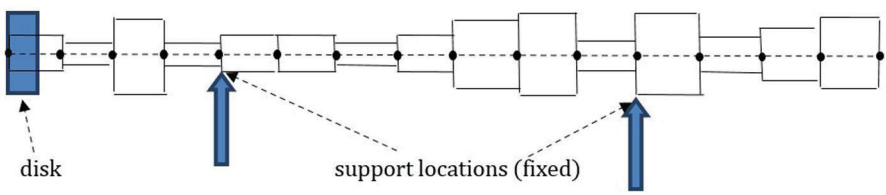


FIGURE 3.23 Optimum shaft geometry by GA – Example 3.8; final optimum solution on shaft diameters that avoids resonance in a specified frequency range.

■

3.4.2 Simulated Annealing (SA)

Simulated annealing (SA) is another evolutionary optimization scheme which is derivative-free and recognized as a probabilistic metaheuristics method like the GA. The name 'simulated annealing' is derived from the fact that the scheme emulates

the process of annealing – evolution of a solid in a heat bath to thermal equilibrium. Annealing is the well-known metallurgical process of heating up a solid and then cooling it slowly until crystallization takes place. At very high temperatures, the atoms of the solid have high energy and as the temperature is reduced in a controlled fashion, the energy reduces, until a state of minimum energy is achieved. Based on this original idea of the SA that dates back to the middle of the 20th century, Metropolis et al. (1953) introduced a path-breaking algorithm** to arrive at the equilibrium of a collection of atoms at a given temperature. This pioneering technique inspired Kirkpatrick et al. (1983) to incorporate it as an optimization tool, wherein temperature serves as a controlling parameter and a measure of diffusion in the evolution of the design variables. Thus, SA begins with a high value of temperature so that the design variables are allowed a wide range of variation. As the algorithm progresses, temperature is made to fall as per a cooling schedule. This often guides the algorithm to a better solution, just as a metal piece achieves a better crystal structure through an actual annealing process.

As temperature decreases, changes are produced yielding successively better solutions and finally giving rise to an optimum set of variables when the temperature is close to zero. The travelling salesman problem (TSP) which is described and solved in Chapter 1 is, in fact, efficiently solved by this thermo-dynamical approach [Cerny 1985].

Description of the basic methodology in SA

SA may be viewed as a sequence of Metropolis algorithms adopted at different values of the controlling parameter T. At each T_k with k denoting an iteration, we use Metropolis algorithm to generate a sequence of trial states, each in the neighbourhood of the current state. With n denoting the design variable space dimension, let $\boldsymbol{x}_k = \left\{x_{i,k}, i = 1,2,\ldots,n\right\}$ and $\hat{\boldsymbol{x}}_k = \left\{\hat{x}_{i,k}, i = 1,2,\ldots,n\right\}$ be the current and a trial state with fitness values $E_k = f\left(\boldsymbol{x}_k\right)$ and $\hat{E}_k = f\left(\hat{\boldsymbol{x}}_k\right)$ simulating the energy levels in an

---

** path-breaking algorithm – the Metropolis algorithm

The Metropolis algorithm is considered one of the top ten algorithms of the 20th century (Dongarra and Sullivan 2000). Metropolis et al. (1953) proposed a method that made simple an otherwise difficult task: simulating the evolution of a solid in a heat bath to thermal equilibrium. The difficulty is in simulating the Boltzmann distribution that the algorithm uses to characterize the equilibrium state at a temperature T. The distribution gives the probability of a solid particle in state x with energy E in the form:

$$\mathbb{f}(x) = \frac{1}{C}\exp\left(-\frac{E}{k_B T}\right) \tag{i}$$

k_B is the Boltzmann constant. C is a normalization constant that renders $\mathbb{f}(.)$ a valid distribution. The fact that C is not known a priori makes the simulation difficult by straight-forward techniques like transformation methods (Papoulis 1991, Roy and Rao 2017). Metropolis algorithm is a Monte Carlo technique based on the theory of Markov chains. It thus belongs to a large class of sampling algorithms known as MCMC techniques.

annealing process. The trial state $\hat{\boldsymbol{x}}_k$ is generated by applying a random perturbation mechanism, which transforms the state $\boldsymbol{x}_k$ to $\hat{\boldsymbol{x}}_k$. If $\hat{E}_k \le E_k$, the state $\hat{\boldsymbol{x}}_k$ is accepted as the current state; otherwise, $\hat{\boldsymbol{x}}_k$ may still be accepted with an acceptance probability α_k. Thus, the method probabilistically decides, depending on the difference between the fitness values E_k and $\hat{E}_k$, whether the current candidate should be replaced by the trial set or not. This is repeated over a large number of steps at the k^{th} iteration. The Metropolis algorithm assumes that the probability of being in a state $\boldsymbol{x}_k$ with energy $E_k = f(\boldsymbol{x}_k)$ at temperature T_k is given by the Boltzmann distribution:

$$\mathbb{f}(x_{i,k}) = \frac{1}{c}\exp\left(-\frac{E_k}{k_B T_k}\right) \tag{3.45}$$

where c is a normalization constant so that $\mathbb{f}(.)$ is a valid distribution. The choice of this distribution is motivated by the fact that the role of the control parameter T is to (a) keep the probability of accepting a trial state high in the earlier stages and (b) force it towards zero asymptotically. In particular, the initial temperature is generally chosen sufficiently high so as to avoid a local minimum. These twin requirements are satisfied by the Boltzmann distribution (Aarts and Korst 1989). Note that, with the normalizing constant c unknown, to simulate and draw realizations from $\mathbb{f}(.)$ is difficult using straight-forward methods. In achieving the task, the Metropolis algorithm uses MC simulation based on the theory of Markov chains. It generates at each temperature a sequence of trial states by MC simulation. The initial states $x_{i,k}$, $i = 1,2,\ldots,n$ at each temperature T_k are simulated, for instance, from a uniform probability distribution:

$$x_{i,k} = x_{i,L} + u\left(x_{i,U} - x_{i,L}\right),\ i = 1,2,\ldots,n,\ u \sim U(0,1) \tag{3.46}$$

The sequence of trial states generated by a small random perturbation over the current states forms a Markov chain at each temperature T. The algorithm considers these states to be realizations of random variables (RVs) $X_{i,k}$, $i = 1,2,\ldots,n$. RVs are denoted by upper case letters. Here the assumption of a Markov chain is valid in that each trial state in the sequence is generated from a neighbourhood of the current state. A Markov chain is characterized by (a) a state space $S \in \mathbb{R}^l$ – the set in which $x_{i,k}$ takes values and (b) a transition probability matrix $\boldsymbol{P}$. l is the number of steps that the algorithm is allowed to take at a temperature T. Each element P_{ij} of $\boldsymbol{P} \in \mathbb{R}^{l\times l}$ stands for the conditional probability $P(\widehat{X}_{i,k} = \hat{x}_{i,k} \mid X_{i,k} = x_{i,k})$ – the probability of a single step transition to $\hat{x}_{i,k}$ from the state $x_{i,k}$. It is easy to see that if the initial probabilities of the states are given by the vector $\boldsymbol{p}_0$, then after, say, m transitions they are:

$$\boldsymbol{p}_m = \boldsymbol{p}_0 \boldsymbol{P}^m \tag{3.47}$$

Metropolis algorithm cleverly uses the distinctive features of a Markov chain – reversibility and existence of a limiting probability distribution – to simulate the final states in the so-called thermal equilibrium at each temperature. A Markov chain, under certain conditions, reaches a unique stationary distribution $\boldsymbol{p}_s$ which is known as the limiting distribution (see Example A3.8 of Appendix 3). The distribution thus satisfies:

$$\boldsymbol{p}_s = \boldsymbol{p}_s \boldsymbol{P}^m \tag{3.48}$$

The Metropolis algorithm takes the Boltzmann distribution $f(.)$ as the limiting distribution $\boldsymbol{p}_s(.)$ of each Markov chain in SA. It next applies the detailed balance condition associated with the reversibility of a Markov chain which is stated as:

$$f(x_{i,k})P_{ij} = f(\hat{x}_{i,k})P_{ji} \tag{3.49}$$

The algorithm determines the acceptance probability (with the normalization constant cancelled) as:

$$\alpha_k = \min\left(1, \frac{f(\hat{x}_{i,k})P_{ji}}{f(x_{i,k})P_{ij}}\right) = \min\left(1, \exp\left(-\frac{\hat{E}_k - E_k}{k_B T_i}\right)\frac{P_{ji}}{P_{ij}}\right) \tag{3.50}$$

The trial state $\hat{\boldsymbol{x}}_k$ is accepted or rejected according to the above acceptance probability. Thus $\hat{\boldsymbol{x}}_k$ is accepted with probability 1 when $\hat{E}_k \le E_k$. Even if $E_k > E_j$, it is still accepted with a probability $\exp\left(-\dfrac{\hat{E}_k - E_k}{k_B T_i}\right)$. Thus the algorithm accepts not only better solutions but also worse solutions with an acceptance probability. The Metropolis algorithmic step is repeated over enough trial states at each T_k before updating the temperature. Table 3.5 details the steps in the SA algorithm. Equation (3.50) indicates that, initially with higher values of T_k, the probability of higher energy levels is higher with the algorithm accepting failures. As the parameter gradually decreases, the acceptance rate of failures also decreases, implying that the algorithm tends to settle around the available best. Theoretically, an asymptotic convergence that is controlled by a so called cooling schedule to an optimal solution is guaranteed (Aarts and Van Laarhoven 1985, Hajek 1988). A proper 'cooling' schedule (typically based on a logarithmic reduction) is needed for the SA algorithm to attain convergence to the desired optimum in finite time.

Example 3.9. We take up the shaft dynamics problem of the previous example and obtain the optimum shaft geometry by SA to avoid resonance in the desired frequency range.

Solution. The initial value of the control parameter *T* is taken as 5 and the annealing schedule is $T_k = 0.8T_j$ where k represents the current iteration and *j* the last one.

TABLE 3.5
Main Steps of the SA Algorithm

Step 1. Let $\boldsymbol{x}_0$ be the initial vector of design variables before the start of iterations. Specify the initial temperature T along with the annealing (cooling) schedule and the acceptance probability criterion. Compute the initial cost (fitness value), $f(\boldsymbol{x}_0)$. Fix the final (minimum) temperature T_f to stop the computations. Fix a suitable limit l on the number of Metropolis algorithmic steps at each temperature. Set the counter for a successful step Isuc=1.

Step 2. Start iterations with current temperature T_k. At the k^{th} iteration, let x_k be the current state; compute $E_k = f(\boldsymbol{x}_k)$.

Step 3. Generate a new set of design variables and get a trial state $\hat{\boldsymbol{x}}_k$ as per Equation (3.46). Compute the new cost $\hat{E}_k = f(\hat{\boldsymbol{x}}_k)$.

Step 4. Find the change in the cost due to the new set, i.e. $\hat{E}_k - E_k$ andw decide on acceptance or rejection of the trial state $\hat{\boldsymbol{x}}_k$ according to the following Metropolis criterion:

i) accept the new cost E_k and replace $\boldsymbol{x}_k$ by $\hat{\boldsymbol{x}}_k$

if $\hat{E}_k - E_k \le 0$ or $\exp\left(-\dfrac{\hat{E}_k - E_k}{k_B T_k}\right) \ge U(0,1)$, a random number from a uniform distribution. Increase the number of successful steps by 1, i.e., Isuc = Isuc + 1.

ii) reject otherwise.

k_B is called the Boltzmann constant and is taken as unity.

Step 5. At the end of the inner steps (= l at a temperature T_k), reduce the temperature according to the annealing schedule:

$T_{k+1} = CT_k, C \in (0,1)$. If the current $T_{k+1} > T_f$, repeat steps 2-5; otherwise end computations.

We refer to the FE model in Figure 3.17b and assume that the shaft is initially uniform with diameter equal to 0.02 m for the initial iteration. With the lower and upper bounds $\boldsymbol{x}_L$ and $\boldsymbol{x}_U$ of the shaft element diameters taken as 0.004 m and 0.05 m, respectively, the trial set $\hat{\boldsymbol{x}}_k$ of diameters is generated at any iteration with the help of Equation 3.46. The final (minimum) temperature to stop the computations is taken as 2.0E-3. The iterative process is carried out at each T with 200 iterations. The results obtained by SA are shown in Figures 3.24a–b.

■

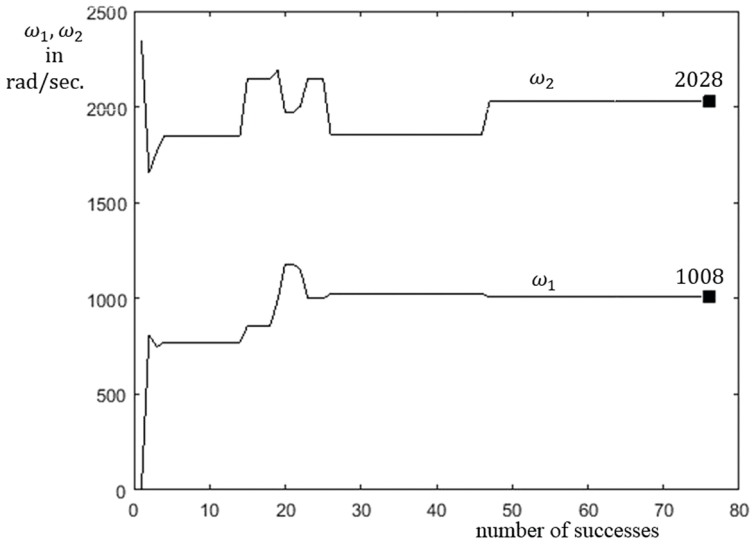


FIGURE 3.24a Optimum shaft geometry (Example 3.8) by SA to avoid resonance in a specified frequency range; evolution of first two natural frequencies ω_1 and ω_2.

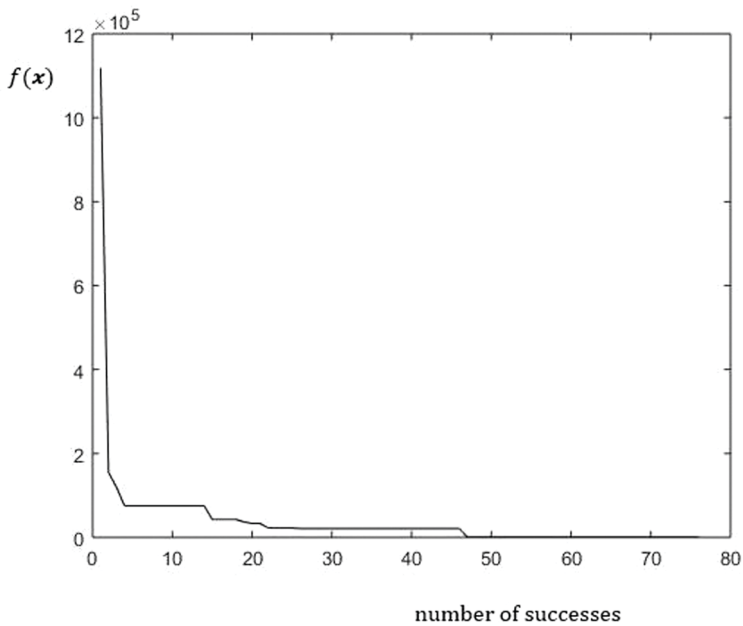


FIGURE 3.24b Optimum shaft geometry (Example 3.8) by SA to avoid resonance in a specified frequency range; evolution of objective function (in Equation 3.44) with number of successes during iterations (finally attaining a minimum value of 848.0).

Note that the annealing schedule of SA is similar to the strategy adopted in GA,

viz. of decreasing the mutation probability as iterations progress. One disadvantage common in stochastic local search algorithms is that the definition of some control parameters (initial temperature, cooling schedule) is somewhat subjective (Wong and Constantinides 1998) and must be done from an empirical basis. This means that, given a problem, the algorithm must be tuned in order to maximize its performance. The diverse applications (Chibante 2010) of SA include inverse problems or parameter identification (Silva Neto and Özişik 1994, Souza et al., 2007) and optimal control systems design (Grimble and Johnson 1988, Ogata 1997).

3.4.3 Particle Swarm Optimization (PSO)

PSO is based on swarm or group intelligence and is a behaviourally (socially) inspired algorithm. PSO, originally proposed by Kennedy and Eberhart (1995), mimics the behaviour of social organisms, e.g. a swarm of insects such as ants, bees and wasps, a flock of birds, school of fish, etc. Each candidate, denoting a bee in a colony or a bird in a flock etc., moves randomly, guided only by its own 'intelligence' and the collective or 'group intelligence' of the swarm. For example, if a candidate finds a good path to food, the rest of the swarm will also be able to follow the path instantly even if their locations are far away. Each candidate, whose motion is characterized by position and velocity, wanders around in the design space and remembers the best position it has discovered so far. The particles (candidates) communicate information on good positions with one another and adjust their individual positions and velocities accordingly.

A unique feature of the PSO is that the search space is explored by a combination of the swarm's previous best and the individuals' previous best positions. At the heart of the swarm behaviour are three driving factors (which may, in part, contradict each other – a reflection of the exploration–exploitation trade-off): (1) cohesion – stick together, (2) adhesion – do not come too close, and (3) alignment – follow the general heading of the flock. To summarize, the PSO is derived following the model below.

(1) When a particle locates a target (i.e. an available extremum of the objective functional), it instantaneously shares the information with all others (non-local interaction across space-separated particles).
(2) All other particles tend to come towards the target.
(3) However, in doing so, each particle exercises its own intelligence consistent with its past memory.

Thus, the model performs a random search, restricted by the conditions above, in the design space so that, with progressing iterations, it should approach the global extremum. The main features of the algorithm are described in Table 3.6.

c_1 and c_2 are also known as the cognitive and social parameters, respectively. Fine tuning of these parameters and the inertia weight w improves the performance of the PSO [Clerc and Kennedy 2002]. While a larger value of the inertia weight may help in global exploration initially, a smaller value applies to local exploration (e.g. near the final stages when much of the design space is already explored). This parameter like the mutation probability in GA thus has functional similarities with the temperature parameter in the SA. Note that many such meta-heuristic optimization schemes

TABLE 3.6
Main Steps in PSO Algorithm

Initialize an n-dimensional search space, and generate N_p particles (defining a swarm of N_p realizations or sets of n design variables).
The j^{th} particle is defined by its n -dimensional position and velocity vectors denoted by

$$\boldsymbol{x}^{[j]} = \left(x_1^{[j]}, x_2^{[j]}, \ldots, x_n^{[j]}\right)^T \text{ and } \boldsymbol{v}^{[j]} = \left(v_1^{[j]}, v_2^{[j]}, \ldots, v_n^{[j]}\right)^T, \quad j = 1,2,\ldots,N_p \text{ respectively.}$$

Evaluate the cost function for N_p particles and start the iteration counter, 1 to k_{max}.

Step 1. At any iteration, say $k \in \left(1, k_{max}\right)$, pick $p_{best}^{[j]}$, the best position of the j^{th} particle and pick $\boldsymbol{g}_{best}$, the best position of the swarm over all the past iterations, i.e. 1 to $k-1$.

Step 2. Update the particles as:

$$\boldsymbol{v}^{[j]}(k+1) = \boldsymbol{w}\boldsymbol{v}^{[j]}(k) + c_1 r_1 \left(\boldsymbol{p}_{best}^{[j]} - \boldsymbol{x}^{[j]}\right) + c_2 r_2 \left(\boldsymbol{g}_{best} - \boldsymbol{x}^{[j]}\right)$$

$$\boldsymbol{x}^{[j]}(k+1) = \boldsymbol{x}^{[j]}(k) + \boldsymbol{v}^{[j]}(k+1) \tag{3.51a,b}$$

$w \in \mathbb{R}^+ -$ inertia weight (or weight parameter on the previous velocity of the particle),

r_1 and r_2 – random numbers uniformly distributed in $[0,1]$ and
c_1 and c_2 – learning factors of a particle - the first from the knowledge of its own success $p_{best}^{[j]}$ and the second from that of the best position $\boldsymbol{g}_{best}$ of the swarm.

Step 3. Repeat steps 1 and 2 till a stopping criterion is satisfied.

typically lack scientific rigour, grounded as they are in intuitive reasoning drawn from social or biological observations.

Example 3.10. We solve the shaft dynamics problem of the Example 3.8 by PSO.

Solution. The objective function is the same as in Equation (3.42) and the initial set of design variables $\boldsymbol{x}_j = x_{L,j}\{\mathbf{1}\}_{N_p \times 1} + \left(x_{U,j} - x_{L,j}\right)\{u\}_{N_p \times 1}, \quad j = 1,2,\ldots,n$ (velocity vector) where $\boldsymbol{x}_L \in \mathbb{R}^n$ and $\boldsymbol{x}_U \in \mathbb{R}^n$ are the specified lower and upper bounds for the design variables. $\{\mathbf{1}\}$ stands for a vector of ones. $u \sim U(0,1)$ is the vector of uniformly distributed random numbers. The weight parameter w is similar to the control parameter T of SA. w is gradually decreased as iterations progress:

$$w_k = w_{max} - \frac{k}{Iter_{max}}\left(w_{max} - w_{min}\right) \tag{3.52}$$

w_{max} and w_{min} are the maximum and minimum values, presently chosen to be 0.9 and 0.4, respectively. $Iter_{max} = 100$ is the maximum number of iterations and k stands for the k^{th} iteration. Parameters c_1 and c_2 are both selected as 2.0. The results are shown in Figures 3.25a–b.

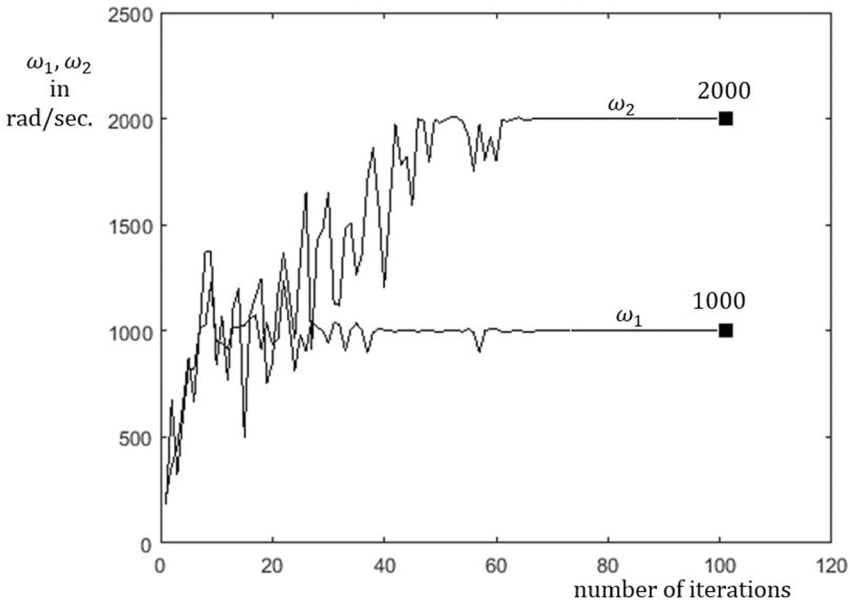


FIGURE 3.25a Optimum shaft geometry (Example 3.8) by PSO to avoid resonance, results with c_1 and c_2 = 2; evolution of the first two natural frequencies ω_1 and ω_2.

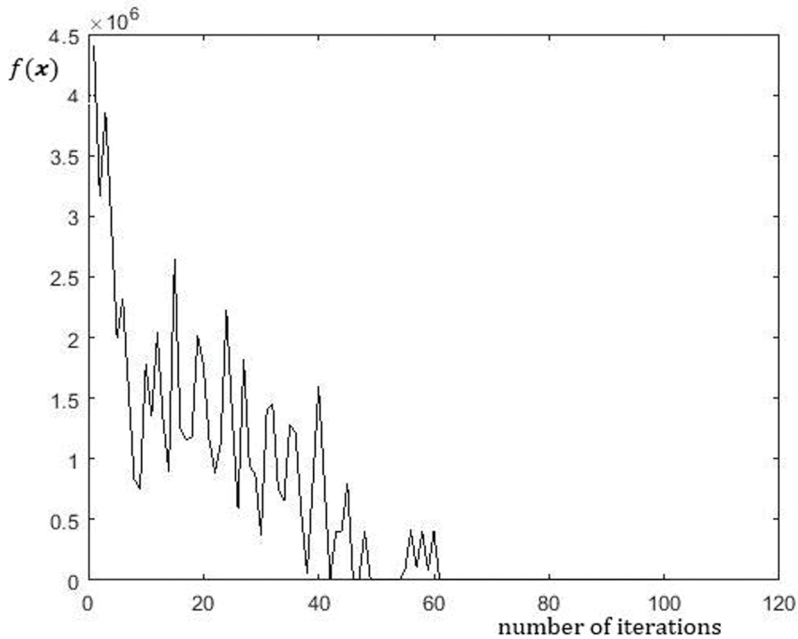


FIGURE 3.25b Optimum shaft geometry (Example 3.8) by PSO to avoid resonance, results with c_1 and c_2 = 2.0; evolution of the objective function (in Equation 3.42) with iterations (finally attaining the minimum value of 0.0).

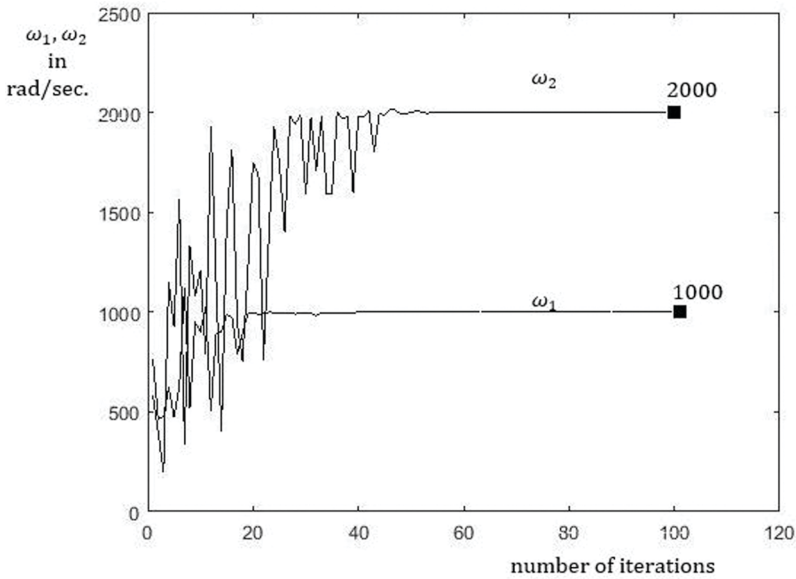


FIGURE 3.26a Optimum shaft geometry (Example 3.8) by PSO to avoid resonance; results with $c_1 = 1$ *and* $c_2 = 2$; evolution of the first two natural frequencies ω_1 and ω_2 in rad/s.

Changes in parameters may have strong influence on the performance of PSO. For instance, the results in Figure 3.25 pertains to the cognitive and social parameters c_1 and c_2 which are both assumed to be 2. If we choose c_1 and c_2 as 1 and 2 respectively, results are shown in Figures 3.26a–b which indicates that the algorithm shows better convergence.

■

Improved versions of the scheme for better performance may be found in Van den Bergh and Engelbrecht (2004), Jiao et al. (2008), and Hu et al. (2012) that include those with adaptivity to change w, c_1 and c_2 for better convergence. The PSO has been extensively used (Engelbrecht 2006) for many scientific and engineering purposes – see Venter and Sobieszczanski-Sobieski (2004) for optimization of transport aircraft wing, Begambre and Laier (2009) and Kang et al. (2012) for damage detection and Luh et al. (2011) for topology optimization. Several approaches to solve constrained optimization problems by the PSO are suggested in Hu and Eberhart (2002) and He and Wang (2007). Interested readers may find a review on swarm intelligence algorithms and their applications in Brezočnik et al. (2018).

3.4.4 Differential Evolution (DiEv)

Differential evolution (DiEv) – another evolutionary optimization strategy – relies on parallel direct search and works with a randomly chosen population. Like the GA, DiEv also has its strategic operations similar to mutation, cross-over and selection.

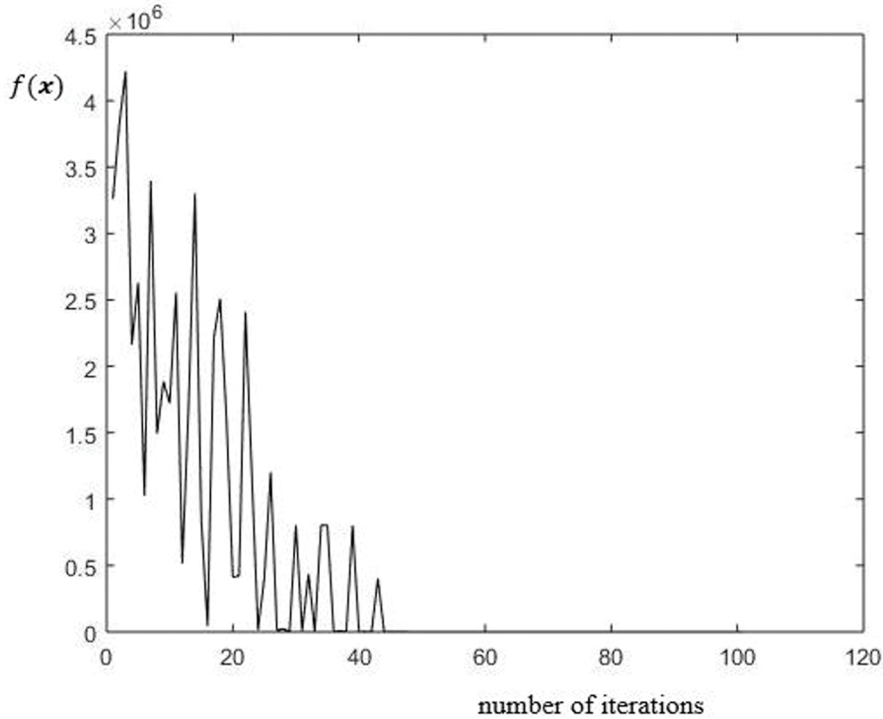


FIGURE 3.26b Optimum shaft geometry (Example 3.8) by PSO to avoid resonance; results with $c_1 = 1$ *and* $c_2 = 2$; evolution of the objective function (in Equation 3.44) with iterations (finally attaining the minimum value of 0.0).

The method starts with an initial set of randomly generated population consisting of N_p parameter vectors. DiEv generates new vectors by mutation which, for this method, is defined as adding the weighted difference between two randomly chosen vectors to a randomly chosen third one (see Figure 3.27). The elements of the mutated vector are then added to those of another randomly chosen vector, the 'target vector', to yield the so-called trial vector. Such an operation is similar to the crossover used in the GA. The trial vector is accepted if it improves the fitness value over the last. This step is the selection. To ensure that all the available vectors take part in the exploration, each vector is forced to assume the role of the target vector once (the random choice of the target is thus in the set of vectors not yet chosen as targets). The algorithm is detailed in Table 3.7.

Equation (3.53) indicates that the mutation operation in GA is replaced by a differential form and hence the name 'differential evolution'. The algorithm in Table 3.7 is a basic version of DiEv and is generally denoted by DiEv/rand/1/bin. '/rand/' indicates that $\boldsymbol{v}_k^{[r_1]}$ in Equation (3.53) is randomly selected for performing mutation. '/1/' indicates the number of difference vectors used during mutation. In Equation (3.53),

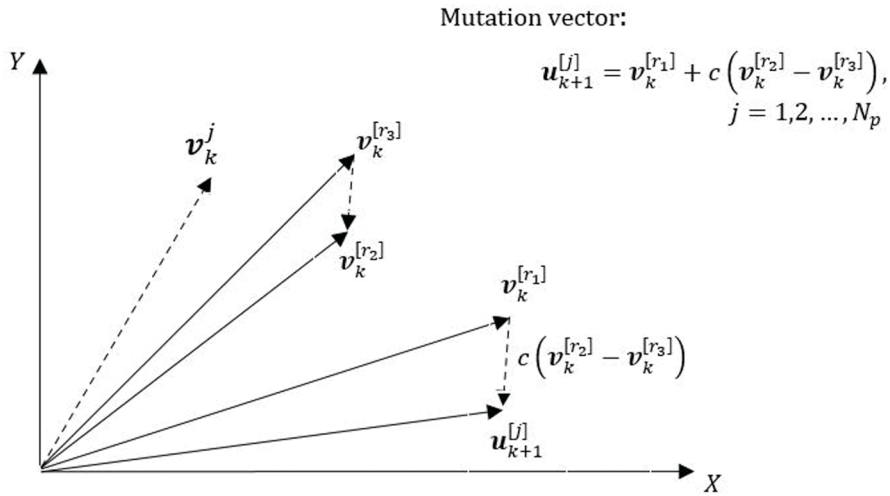


FIGURE 3.27 Mutation operation in DiEv at the end of the k^{th} iteration in a two-dimensional parameter space (for details on notations, see Table 3.7).

only one difference vector $\left(\boldsymbol{v}_k^{[r_2]} - \boldsymbol{v}_k^{[r_3]}\right)$ is used. '/bin/' indicates that independent binomial experiments decide crossover in Equation (3.53). A typical variant may be DiEv/best/2/bin wherein mutation is executed over the best population vector, i.e.:

$$\boldsymbol{u}_{k+1}^{[j]} = \boldsymbol{v}^{[best](k)} + c_1\left(\boldsymbol{v}_k^{[r_1]} - \boldsymbol{v}_k^{[r_2]}\right) + c_2\left(\boldsymbol{v}_k^{[r_3]} - \boldsymbol{v}_k^{[r_4]}\right), j = 1,2,\ldots,N_p \qquad (3.56)$$

and two difference vectors are involved in mutation. $\boldsymbol{v}_k^{[best]}$ is the best population vector (based on the fitness value) at the k^{th} iteration among $\boldsymbol{v}_k^{[j]}$, $j = 1,2,\ldots,N_p$. Other, possibly more efficacious, variants of DiEv have been reported [Wang et al. 2011], differing from the basic version of Table 3.7 by way of how mutation and crossover are performed.

Example 3.11. We solve the shaft dynamics problem of Example 3.8 by DiEv.

Solution. Referring to the last example by PSO, we similarly choose the initial set of design variables as $\boldsymbol{x}_j = x_{L,j}\{1\}_{N_p\times 1} + \left(x_{U,j} - x_{L,j}\right)\{u\}_{N_p\times 1}$, $j = 1,2,\ldots,n$. The population size N_p is 15. The results are in Figures 3.28a–b.

TABLE 3.7
Salient Features of the DiEv Algorithm

Initialize N_p, the population size in a specified n-dimensional parameter space. Let the cost function be $f(x)$. Randomly generate the initial set of N_p parameter vectors, $v_0^{[j]}$, $j = 1,2,\ldots,N_p$, each of size n, whilst ensuring that the vector elements are within the prescribed limits of the design space and reasonably well scattered. Start iterations $k = 1,2,\ldots,k_{max}$.

Step 1. Generate the new set of N_p vectors by mutation:

$$\boldsymbol{u}_{k+1}^{[j]} = \boldsymbol{v}_k^{[r_1]} + c\left(\boldsymbol{v}_k^{[r_2]} - \boldsymbol{v}_k^{[r_3]}\right),\ j = 1,2,\ldots,N_p \tag{3.53}$$

r_1, r_2 and r_3 are random integers (different from each other) drawn from a discrete uniform distribution on [1: N_p] and $c \sim U(0,1)$.

Step 2. Perform cross-over to get the trial vector:

$$\begin{aligned} \boldsymbol{t}_{(k+1),i}^{[j]} &= \boldsymbol{u}_{(k+1),i}^{[j]} \text{ if } p_i \le q \text{ or } i = \varsigma_{rand} \\ &= \boldsymbol{v}_{k,i}^{[j]} \text{ otherwise, } j = 1,2,\ldots,N_p, i = 1,2,\ldots,n \end{aligned} \tag{3.54}$$

where $p_i \in U\ [0,1]$ and q a user-specified real number in $[0,1]$. ς_{rand} is a randomly chosen integer in $[1,N]$. The operation ensures that at least one element $u_{(k+1),i}^{[j]}$ (denoted by the subscript 'i') enters the trial vector of each j^{th} particle.

Step 3. Selection: the final vectors at the $(k+1)^{th}$ iteration are determined as:

$$\begin{aligned} \boldsymbol{v}_{k+1}^{[j]} &= \boldsymbol{t}_{k+1}^{[j]} \text{ if } \boldsymbol{t}_{k+1}^{[j]} \\ \boldsymbol{v}_{k+1}^{[j]} &= \boldsymbol{v}_k^{[j]} \text{ otherwise} \end{aligned} \tag{3.55}$$

Step 4. Repeat steps 1-3 till a stopping criterion is satisfied.

■

While the selection of N_p, c and q is problem-dependent, a possible guidance may be $5n \le N_p \le 10n$, $c = 0.5$ and $q = 0.1$. Different population vector strategies and control parameter settings with possible adaptivity have been studied (Fan and Lampinen 2003, Brest et al. 2006, Mallipeddi and Suganthan 2008) for improving the performance of DiEv with respect to many benchmark functions that include multi-modal and hybrid composite functions.

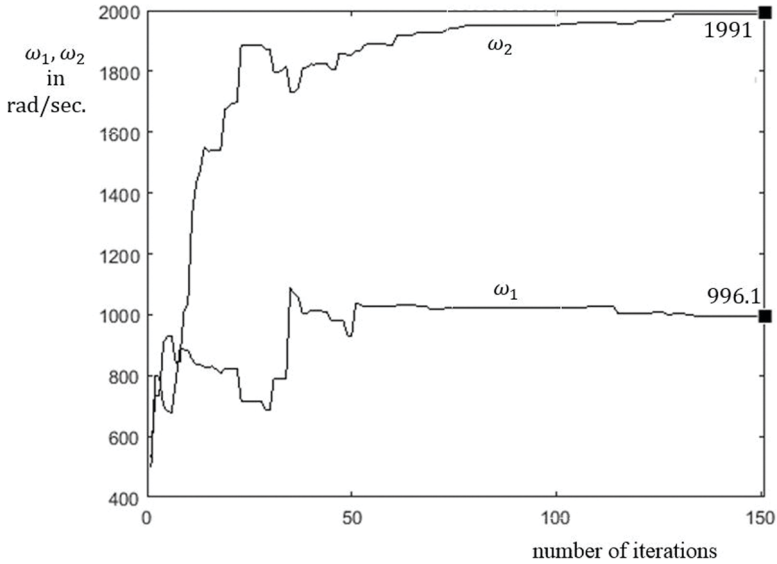


FIGURE 3.28a Optimum shaft geometry (Example 3.8) by DiEv to avoid resonance; N_p =15, $q = 0.1$; evolutions of the first two natural frequencies ω_1 and ω_2.

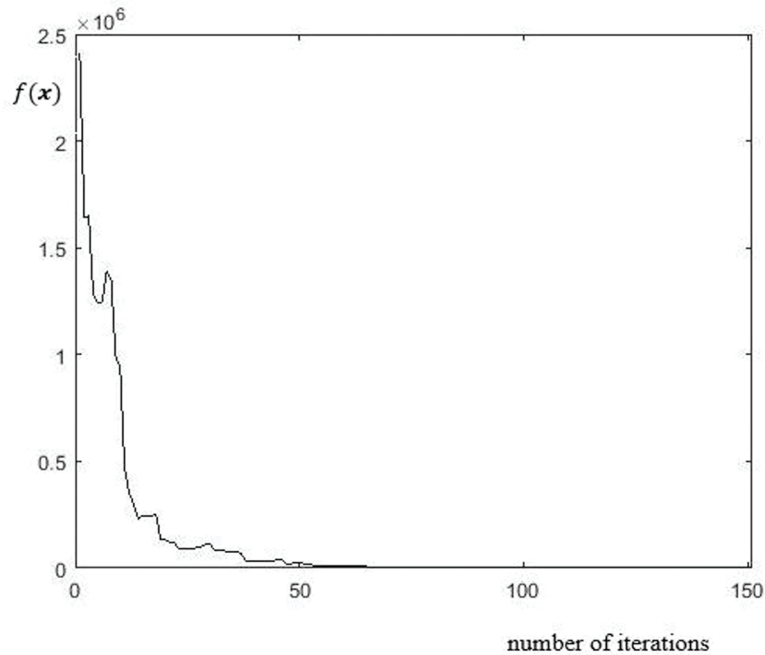


FIGURE 3.28b Optimum shaft geometry (Example 3.8) by DiEv to avoid resonance; N_p = 15, $q = 0.1$; evolution of the objective function with iterations (finally attaining a minimum value of 98.02).

CONCLUDING REMARKS

This chapter has outlined a class of derivative-free yet reliable optimization techniques commonly known by the name "direct search methods". Vis-à-vis the quasi-newton methods, these schemes are computationally inexpensive. Treated as a seemingly hodgepodge collection of algorithms based on heuristics during the early seventies, these methods became increasingly popular as their global convergence behaviour got well established in recent times. Among these, the method of pattern search by Hooke and Jeeves and of simplex by Nelder and Mead occupy a prime place. They are followed in this chapter by the methods of Powell and Rosenbrock. Without the need of derivatives, both Powell and Rosenbrock have shown, although by different approaches, how one-dimensional searches during iterations could be exploited to obtain information on the curvature of the objective function and thus accelerate the search. Trust region method is yet another robust technique discussed in the chapter. The method uses at each iteration a local submodel with a local approximation to the original objective function in a trust region. The trust region size is updated as per a certain merit function and iterations are continued with no need for derivative computations. Another significant aspect of the chapter is the narration of a few popular evolutionary optimization methods – genetic algorithm, simulated annealing, particle swarm optimization and differential evolution – which are again of heuristic/meta-heuristic origin and derivative-free. Underlying each of these methods, there is a probability model to iteratively sample and update the solution. This is perhaps why they often go by the name 'stochastic search methods'. One can expect that the notion of random sampling and evolution of possible solutions should efficiently explore the search space, though at the cost of possibly slower convergence. Be that as it may, the popular adoption of these schemes is not only due to algorithmic simplicity, but mainly because of their effectiveness in treating many NP-hard (Appendix 1) optimization problems. Even though these derivative-free techniques primarily aim at solving unconstrained optimization problems, they may be hybridized with any of the methods such as Lagrange multipliers, augmented Lagrangian or penalty function of Chapter 2 to handle problems with constraints.

Despite a wide adoption of evolutionary methods of heuristic/meta-heuristic origin, the underlying justification is often based on sociological or biological metaphors that are hardly founded on a sound probabilistic basis even though a random search forms a key ingredient of each of the algorithms. An exception may be the method of simulated annealing (SA). The last method may seem to adopt the simple notion of probability, like other evolutionary schemes, whilst updating the current state. However, its working requires a deeper understanding of MCMC methods which are in turn based on the theory of Markov chains (Appendix 3). The Metropolis algorithm of SA which is cited as one of the top ten algorithms of the 20th century, was originally designed to emulate the metallurgical process of annealing using the theory of Markov chains. The algorithm is cleverly exploited in SA as an optimization tool with temperature as a controlling parameter. The rest of this book is devoted to Riemannian geometric variants of optimization methods in both deterministic and stochastic settings.

EXERCISES

1. The mean-variance portfolio theory of Markowitz (1952) is the basic model in finance for portfolio selection. In this model, each asset (or stock) is characterized by its return which is a random variable. It obviously carries with it, a risk measured in terms of variance (volatility) of its return. Let R_i represent the expected return of the i_{th} asset and $\boldsymbol{C}$, the covariance matrix with each of its elements C_{ij} denoting the covariance between the i_{th} and j_{th} assets. The model as formulated by Markowitz envisages minimum risk while achieving a pre-specified expected return, say, R. Thus, it is a constrained quadratic minimization problem where it is required to minimize the risk, i.e.:

$$\text{minimize: } \sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij}x_i x_j$$

$$\text{s. t. } \sum_{i=1}^{N} R_i x_i = Q, \sum_{i=1}^{N} x_i = 1 \text{ and } x_i, i = 1,2,\ldots,N > 0, \qquad \text{(E3.1a,b)}$$

Here, $x_i, i = 1,2,\ldots,N$ are the weights (proportions of total investment) of the chosen stocks. The constraint $\sum_{i=1}^{N} x_i = 1$ indeed requires cent percent investment in the portfolio. Find the minimum variance portfolio by HJ and NM methods for a typical portfolio selection with $N = 3$, $\boldsymbol{C} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}$, $\boldsymbol{R} = \begin{pmatrix} 0.4 & 0.4 & 0.8 \end{pmatrix}$ and $Q = 3$.

[**Hint:** Use Lagrange multipliers]

2. This problem is to highlight the application of approximating a continuous probability distribution (target pdf) by a mixture of normal pdfs. The applications include varied disciplines such as astronomy (Newcomb 1886), biology (Niemi 2009) and finance (McLachlan and Peel 2000, Norets and Pelenis 2011).

In *n*-dimensional normal mixture model, the mixture density is expressed as $\sum_{i=1}^{n} p_i \; N\left(m_i, \sigma_i^2\right)$ where p_i is the weight associated with i^{th} normal component while m_i *and* σ_i^2 are its mean and variance. Note that $\sum_{i=1}^{n} p_i = 1$. Thus, if $\boldsymbol{\theta} = \left(\boldsymbol{p}, \boldsymbol{m}, \boldsymbol{\sigma}^2\right)^T$ represents the vector of unknown parameters in the model, it is of size $3n$ with $\boldsymbol{p} \in \mathbb{R}^{n-1}$ *and* $\boldsymbol{m}$, $\boldsymbol{\sigma}^2 \in \mathbb{R}^n$. Given the target *pdf* $\boldsymbol{f_Z(z)}$, it is required to estimate $\boldsymbol{\theta}$ so that the true observations $z = \left\{z_1, z_2, \ldots, z_N\right\}^T$ are most likely to have been realized from the assumed normal mixture pdf.

Task A: If the method of maximum likelihood estimation (MLE) – Section 3.2.2 – is adopted, the problem involves the following minimization problem given by:

$$\text{minimize } l(\boldsymbol{\theta};z)=\sum_{j=1}^{N}\log\left(\sum_{i=1}^{n}\left(\frac{p_i}{\sqrt{2\pi\sigma_i^2}}\exp\left(-\frac{1}{2\sigma_i^2}\left(z_j-m_i\right)^2\right)\right)\right) \tag{E3.2}$$

Suppose that the target pdf $\not{f}_Z(z)$ is the log χ^2 (log-Chisquare) pdf with one degree of freedom (Papoulis 1991) which is given by:

$$\not{f}_Z(z)=\frac{1}{\sqrt{2}\,\Gamma\left(\frac{1}{2}\right)}\,z^{-1/2}\exp\left(-\frac{z}{2}\right),\quad 0\le x\le\infty \tag{E3.3}$$

Estimate the parameters $\boldsymbol{\theta}$ in the normal mixture model by MLE using HJ method of optimization. (**Note**: results are given in Figure E3.1.)

Task B: MLE requires the likelihood function to be bounded over the parameter space which may not be satisfied by the discrete normal mixture model. An alternative approach is to estimate by moment generating function (MGF)†† method (Quandt and Ramsey 1978, Schmidt 1982). This approach minimizes the sum of squares of the distance between the 'sampling' MGF and its empirical counterpart. The 'sampling' MGF is from the assumed normal mixture model:

$$\Phi(s,\boldsymbol{\theta})=E\left[e^{sZ}\right]=\sum_{i=1}^{n}p_i\left\{\exp\left(m_i s+\frac{1}{2}\sigma_i^2 s^2\right)\right\} \tag{E3.4}$$

Form the available observations, the empirical (theoretical) MGF is:

$$\hat{\Phi}(s,z)=\frac{1}{N}\sum_{j=1}^{N}\exp\left(sz_j\right) \tag{E3.5}$$

---

†† Moment generating function

If X is a random variable with pdf, moment generating function is defined by:

$$\phi(s)=E[e^{sX}]=\int_{-\infty}^{\infty}e^{sx}\not{f}_X(x)dx \tag{i}$$

For example, for $X\sim N(m,\sigma)$, $\Phi(s)$ is given by:

$$\Phi(s)=\exp\left(ms+\frac{1}{2}\sigma^2 s^2\right) \tag{ii}$$

If s is changed to $j\omega$ (pure imaginary number), one obtains the characteristic function of X:

$$\phi(\omega)=E\left[e^{j\omega X}\right]=\int_{-\infty}^{\infty}e^{j\omega x}\not{f}_X(x)dx \tag{iii}$$

Differentiating (i) 'n' times, one obtains:

$$\frac{d^n\Phi(s)}{ds^n}=E\left[X^n e^{sX}\right]\Rightarrow\frac{d^n\Phi(0)}{ds^n}=E\left[X^n\right]=E\left[X\right]^n \tag{iv}$$

Thus, the derivatives of $\Phi(s)$ at $s=0$ yields the moments of the random variable X and hence the justifies the name 'moment generating function'.

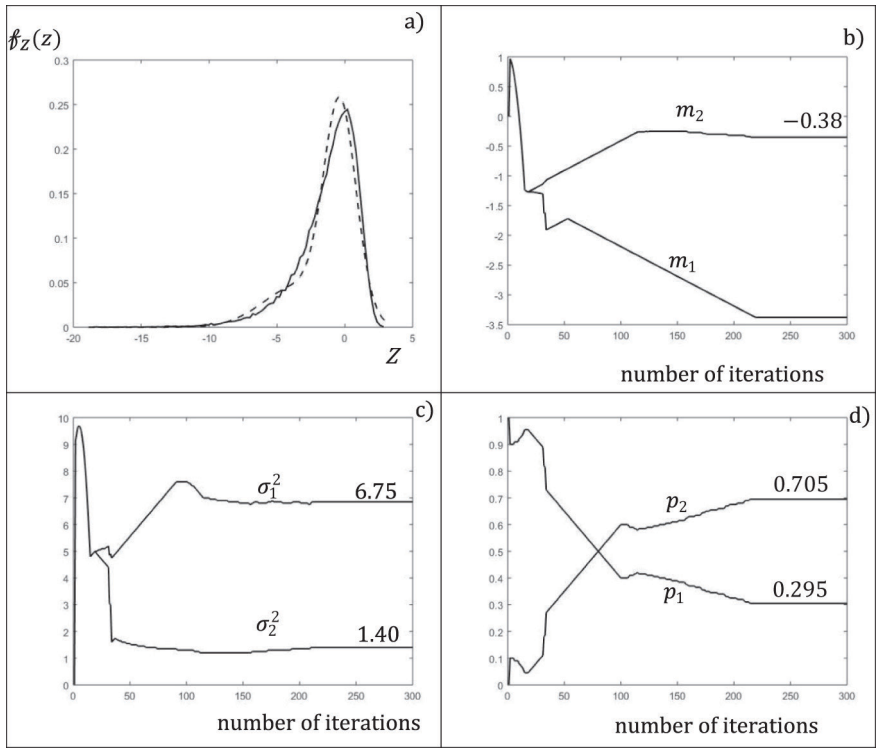


FIGURE E3.1a–d Result for Task A: Optimization by HJ method; MLE estimate of parameters in normal mixture model approximating log χ^2 (log-Chi square) pdf, $N = 50000$ and $n = 2$: (a) original pdf – in dark line and approximated pdf – in dashed line, (b) means m_1 and m_2, (c) variances σ_1^2 and σ_2^2 and (d) weights p_1 and p_2.

With a set of grid points $s_1, s_2, \ldots s_M$ chosen, one needs to minimize the error between the empirical and 'sampling' MGFs, i.e.:

$$\text{minimize: } e(\boldsymbol{\theta}) = \sum_{k=1}^{M} \left(\widehat{\Phi}(s_k, \boldsymbol{z}) - \Phi(s_k, \boldsymbol{\theta}) \right)^2 \tag{E3.6}$$

The number of grid points M may be chosen to be equal to the number of unknown parameters in the normal mixture model.

Hint for Task B: Solve the optimization problem in (E3.6) using any of the derivative-free methods to estimate the parameters $\boldsymbol{\theta}$ by MGF approach.

3. Minimize the function $f(x) = 2x_1^3 + 4x_1x_2^3 - 10x_1x_2 + x_2^2$ (Ravindran et al. 2006) by Rosenbrock's rotating coordinates and Powell's conjugate direction methods.

4. Minimize the Camelback function (Molga and Smutnicki 2005):

$$f(x,y)=\left(4-2.1x^2+x^{4/3}\right)x^2+xy+\left(-4+4y^2\right)y^2$$

$$\text{s. t.}-1.5\le x\le 1.5,-2\le y\le 2 \qquad \text{(E3.7a,b)}$$

by Rosenbrock's rotating coordinates, Powell's conjugate direction and Trust region methods.

5. Solve for weight minimization by GA, of the 10-member plane truss in Figure 2.12, Chapter 2. Also see Example 3.1. Consider an additional constraint on member stresses with allowable stress being 9.5 KN/sq. cm.

6. Consider Cauchy pdf $\mathbb{f}_X(x;\alpha)=\frac{\beta}{\pi(\beta^2+(x-\alpha)^2)},-\infty\le x\le\infty,-\infty\le\alpha\le\infty,$ with β fixed at 0.1. Assuming that samples x, i = 1, 2, …, N are available, find the estimate, by simulated annealing method, of α by maximum likelihood estimation (MLE) – Section 3.2.2. [**Hint**: The log-likelihood function $\sum_{i=1}^{N}\log\mathbb{f}_{X_i}(x_i;\alpha)$. Try the estimation for different $N=1000$ to 10000 in steps of 1000 .]

7. Figure E3.2 shows a quarter model of a vehicle underground excitation $g(t)$. The vehicle dynamics is governed by the following equations of motion:

$$m_1\ddot{z}_1(t)+c_1\left(\dot{z}_1(t)-\dot{z}_2(t)\right)+k_1\left(z_1(t)-z_2(t)\right)=-m_1\ddot{g}$$

$$m_2\ddot{z}_2(t)+c_1\left(\dot{z}_2(t)-\dot{z}_1(t)\right)+c_2\dot{z}_2(t)+k_1z_2(t)=-m_2\ddot{g} \qquad \text{(E3.8a,b)}$$

Here $z_1(t)=x_1(t)-g(t)$ and $z_2(t)=x_2(t)-g(t)$ are relative displacements at the two mass points m_1 and m_2.

The ground excitation due to the road undulations is assumed to be a harmonic signal $A\sin\lambda t$ with A $=0.025$ m and $\lambda=10\frac{rad}{s}$. It is required to minimize the transmissibility of the ground motion to the sprung mass level. This is expressible in terms of the ratio of the amplitude x_1 to the amplitude of the ground excitation A. The passenger comfort is usually expressed as the ratio of the rate of change of vertical acceleration at the sprung mass m_1 to the amplitude of the ground excitation, i.e. $\frac{\dddot{x}_1}{A}$. The maximum jerk experienced at this sprung mass level needs to be restrained to $18\frac{m}{s^3}$. If the design variables are the stiffness k_1 and damping c_1 of the suspension system, one has the optimization problem as:

$$\text{minimize}-\frac{\max\limits_{t\in[0,T]}\left|x_1(t)\right|}{A}$$

$$\text{s. t.}\quad \max_{t\in[0,T]}\ddot{x}_1(t)\le 18\frac{m}{s^3},0\le k_1\le 2E4\ N/m\ ,\ 0\le c_1\le 3000N-s/m \qquad \text{(E3.9a,b)}$$

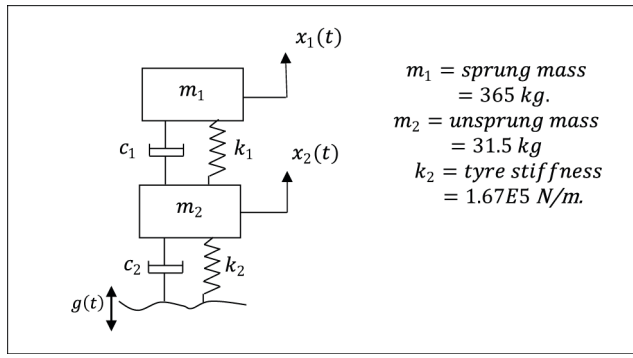


FIGURE E3.2 Quarter car model of a vehicle.

Use any of the derivative-free methods to obtain the optimal solution for maximum passenger comfort.

8. A single degree-of-freedom oscillator is a mass-stiffness-damper system (Figure A3.11, Appendix 3) governed by the second-order ODE: $\ddot{x}(t)+c\dot{x}(t)+kx(t)=p(t)$. When the excitation $p(t)$ is absent, the system undergoes damped oscillations under initial condition disturbance – see Figure E3.3.

The task is: given such a signature of the oscillator in terms of $x(t)$ and $\dot{x}(t)$, it is required to identify the parameters c and k. It is a system identification problem. With $\boldsymbol{x}=(c,k)^T$ as the vector of design variables, it involves error minimization over the time interval $[t_0,t_f]$ of the time histories of $x(t)$ and $\dot{x}(t)$ between the reference solution and solution obtained at each iteration by estimated parameters, i.e.:

$$\text{minimize: } f(\boldsymbol{x})=\sum_{t_0}^{t_f}\left\{\left(x^{(r)}(t_i)-x_k(t_i)\right)^2+\left(\dot{x}^{(r)}(t_i)-\dot{x}_k(t_i)\right)^2\right\}$$

$$\text{s. t. } k_L\le k\le k_U,\ c_L\le c\le c_U,\ \text{and. } k,c\ge 0 \qquad \text{(E3.10a,b)}$$

$\boldsymbol{x}(t)=\left(x(t),\dot{x}(t)\right)^T$. Solve the problem by PSO and DiEV methods. [**Hint**: One may obtain the parameters identified by PSO as shown in Figure E3.4.]

9. Solve the travelling salesman problem (TSP) by the method of Tabu search (Glover and Laguna.1997). The problem is earlier solved in Section 1.3.3, Chapter 1, by search technique using the Metropolis-Hastings algorithm which is identical to the simulated annealing approach.

Tabu search is an exploratory technique like GA, SA and other derivative-free evolutionary algorithms. In Tabu search, all the search moves are investigated until the best solution that is not 'tabu' is achieved. Implementation-wise, it maintains a short-term memory with a list of 'tabu' candidates. Whenever a local optimum is reached, the method opts out of the optimum to search in a new direction with an increase in the cost function by the smallest value. Here resemblance to Metropolis algorithm is

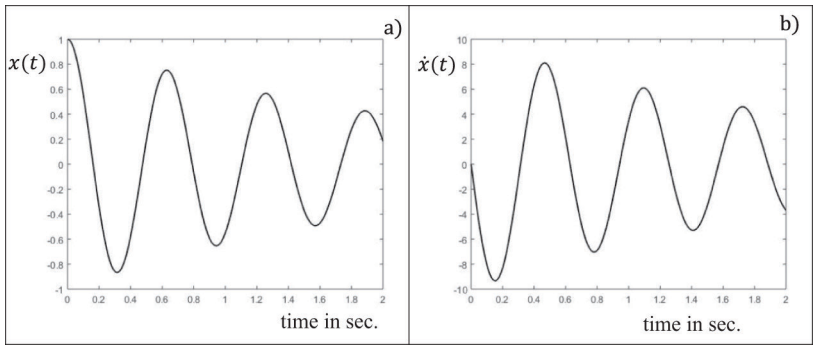


FIGURE E3.3 Reference solution of $x^{(r)}(t)$ and $\dot{x}^{(r)}(t)$ obtained by Runge-Kutta algorithm with true parameters of the system: stiffness $= 100$ N/m and damping coefficient $= 0.9$ N-s/m over the interval $[0,2]$ s.: (a) $x(t)$ and (b) $\dot{x}(t)$.

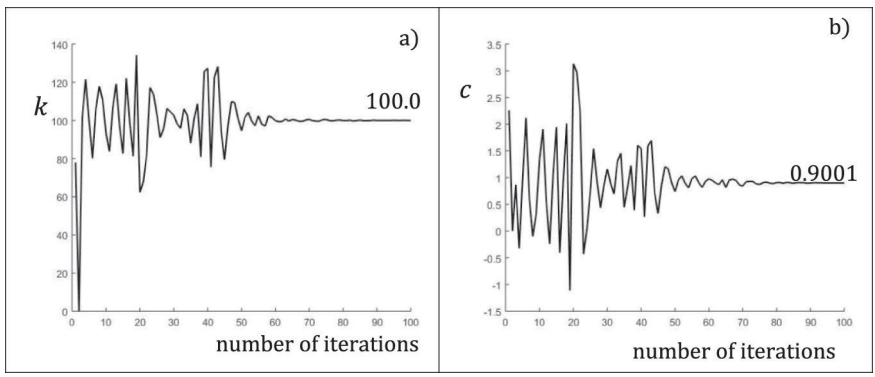


FIGURE E3.4 Error minimization by PSO – a system identification problem for an SDOF oscillator; $k_L = 5, k_U = 150$ and $c_L = 0.1, c_U = 5.0$: (a) evolution of the stiffness parameter k with iterations (finally reaching a value of 100.0) and (b) evolution of the damping parameter c with iterations (finally reaching a value of 0.9001).

noticeable. It ensures to avoid a return to the same optimum and to try for maximizing new information. It is also characterized by search intensification and diversification techniques (Bland 1994, Connor and Tilley 1998).

[**Hint:** Optimum solution by Tabu search technique is shown in Figure E3.5.]

10. Use covariance matrix adaptation (CMA) method (Hansen 2007) to find the optimum of Rosenbrock function $f(\boldsymbol{x}) = 100\left(x_1^2 - x_2\right)^2 + \left(x_1 - 1\right)^2$, $x = \left(x_1, x_2\right)^T$.

[**Hint:** CMA is also an evolutionary optimization algorithm (Hansen 2007). It is a derivative-free method and known to be a robust local search performer. The basic idea in this method is to generate particles at each k^{th} iteration by sampling a

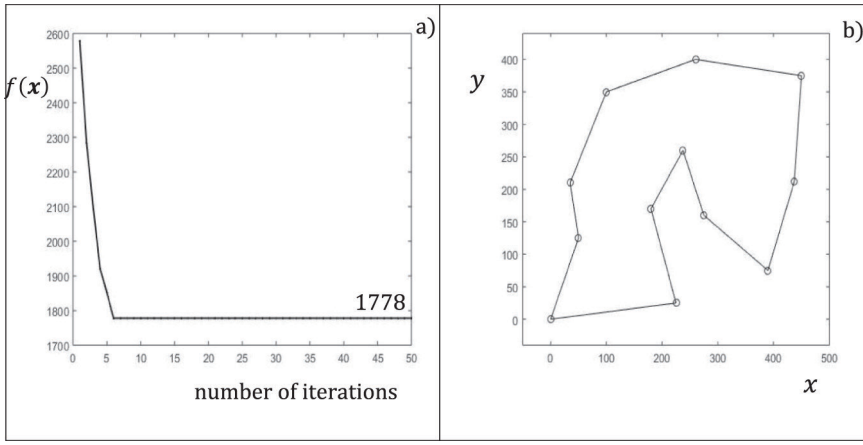


FIGURE E3.5 Solution to travelling salesman problem (TSP) by Tabu search: (a) evolution of the objective function with iterations (finally attaining a minimum value of 1778 units) and (b) optimum tour – the shortest Hamiltonian cycle.

multivariate normal distribution $\mathcal{N}(\mathfrak{M}_k, \boldsymbol{\sigma}_k)$ where $\mathfrak{M}_k \in \mathbb{R}^m$ is the vector of mean values and $\boldsymbol{\sigma}_k^2$, the covariance matrix of the vector $\boldsymbol{X}_k^{[j]} \in \mathbb{R}^m, j = 1.2,..N_p$ of design variables. m is the dimensionality of the problem and N_p the number of particles. At each iteration, the scheme updates the particles as:

$$\boldsymbol{X}_{k+1} \sim \mathfrak{M}_k + s_k \mathcal{N}\left(\mathbf{0}, \mathcal{C}_k^{\frac{1}{2}}\right) \tag{E3.11}$$

$\mathcal{C}_k = \boldsymbol{\sigma}_k^2 \in \mathbb{R}^{m \times m}$ and s_k is known as an 'overall standard deviation' or the step size at k^{th} iteration. $\mathbf{0} \in \mathbb{R}^m$ is the zero-mean vector. Equation (E3.11) is equivalent to sampling $\boldsymbol{X}_{k+1} \sim \mathcal{N}\left(\mathfrak{M}_k, s_k \mathcal{C}_k^{\frac{1}{2}}\right)$. While $\mathfrak{M}_k$ is a weighted average of the N_p particles from the sample $\boldsymbol{X}_k^{[j]}, j = 1.2,..N_p$, $\mathcal{C}_k$ is updated using the covariance information at the current as well as previous step. The updating strategy is equivalent to adopting a quadratic model of the objective function at each search point in the parameter space similar to the approximation of the inverse Hessian matrix in quasi-Newton methods like DFP (Section 2.3.1, Chapter 2).

The solution to the given optimization problem by CMA is shown in Figure E3.6]

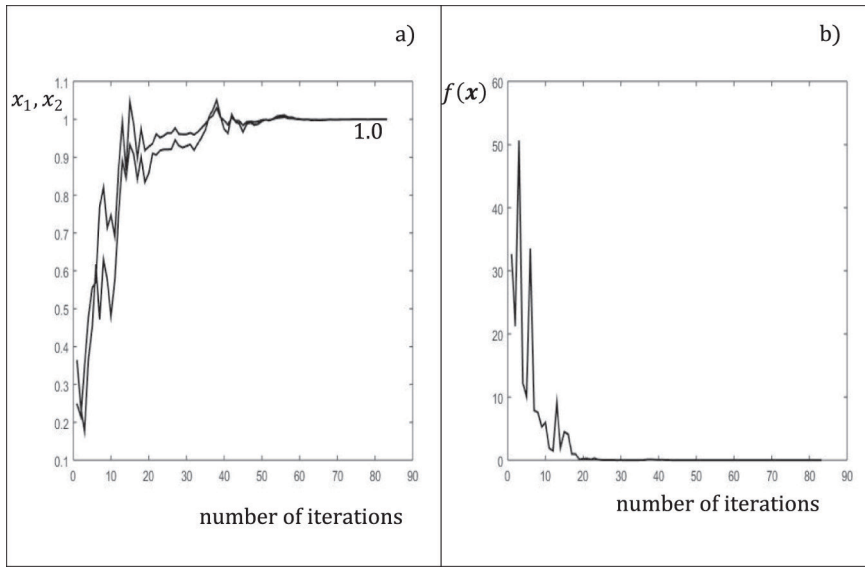


FIGURE E3.6 Optimization of Rosenbrock function by the CMA: $N_p = 6$: (a) evolutions of x_1 and x_2 *with* the optimum $x^* = (1,1)^T$, (b) evolution of the objective function with iterations (finally a minimum value of 1.979E-9).

NOTATIONS

$A_i, i = 1,2,\ldots$	vector of cross-sectional areas (design variables)
A_l and A_u	specified lower and upper bounds for the areas of cross-section
$\boldsymbol{A}_i^k, \boldsymbol{B}_i^k, i = 1,2,\ldots$	vectors in rotating coordinates method of Rosenbrock (Gram-Schmidt procedure)
$\boldsymbol{b}$	a vector
c	a real constant
c_1 and c_2	learning factors of a particle (in PSO)
$\boldsymbol{C}_i$	j^{th} off-spring candidate in GA
$\boldsymbol{d}_i, i = 1,2,\ldots$	coordinate directions (Powell's method of conjugate directions)
$\overline{\boldsymbol{d}}_i, i = 0,1,\ldots$	$\boldsymbol{Q}$-conjugate orthogonal directions in Powell's method of conjugate directions
$\boldsymbol{d}_i^k, i = 1,2,\ldots$	unit orthogonal vectors during the k^{th} stage in rotating coordinates method of Rosenbrock
E	Young's modulus of elasticity
E_k	fitness value (in SA)

$f(\boldsymbol{x})$	Objective function of the design variables x
$\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})$	joint pdf of the vector RV Z in Example 3.2
$\mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})$	marginal pdf of the RV Z_i
$\mathbb{F}_Z(z;\boldsymbol{\theta})$	joint CDF of the vector RV Z in Example 3.2
$\boldsymbol{g}_{best}$	best position of the swarm over all the past iterations (in PSO)
$\boldsymbol{H}(.)$	Hessian matrix
$\boldsymbol{I}_n(\boldsymbol{\theta})$	Fisher information matrix (FIM)
$\boldsymbol{J}(.)$	Jacobian matrix
k_B	Boltzmann constant
$\boldsymbol{K}$	assembled stiffness matrix
$\boldsymbol{K}^e$	element stiffness matrix
$l(\boldsymbol{\theta};z)$	log-likelihood function (Example 3.2)
$L(\boldsymbol{\theta};z)$	likelihood function (Example 3.2)
$\boldsymbol{M}$	assembled mass matrix
$\boldsymbol{M}^e$	element mass matrix
N_e	number of elements in a finite element (FE) model
N_p	size of population
$p_i \in U(0,1)$	uniformly distributed random number in [0,1] (Table 3.7)
p_0	vector of initial probabilities
p_m	vector of probabilities after m transitions
$\boldsymbol{P}$	transition probability matrix
P_i	i^{th} parent candidate (chromosome) in GA
$p_{best}^{[j]}$	best position of the j^{th} particle (in PSO)
$p_{m,k}$	mutation probability (in GA) at k^{th} iteration
$\boldsymbol{\mathcal{P}}(t)$	vector of nodal forces
q	user-specified real number in $[0,1]$ (Table 3.7)
$q(x)$	objective function in trust region sub-program (Section 3.3.3)
$\boldsymbol{Q}$	a matrix
$r_k \in \mathbb{R}$	penalty parameter in the interior penalty method
$r_i(\boldsymbol{x}), i = 1,2,\ldots$	error residuals (trust region method)
$r,\ r_1,\ r_2$	uniformly distributed random numbers in $[0,1]$

R_k	a parameter in trust region method
s	step size (HJ method)
$s(\boldsymbol{\theta};\boldsymbol{Z})$	score function (Equation 3.17)
s_k	step size at the k^{th} iteration
$\bar{s}_i, i=0,1,\ldots$	step size (Section 3.3.2)
$s_i^k, i=1,2,\ldots$	step size for each direction (Section 3.3.1)
$t_i^k, i=1,2,\ldots$	coefficients (Equation 3.24)
$\boldsymbol{t}_{(k),i}^{[j]}$	trial vector obtained after cross-over in DiEv (Equation 3.54)
T_k	temperature parameter (in SA)
$u \sim U(0,1)$	uniformly distributed random number in [0,1]
$\boldsymbol{u}_k^{[j]}$	updated vector of n_j^{th} particle (in DiEv) at the k^{th} iteration (Equation 3.53)
$\boldsymbol{v}^{[j]}$	velocity vector of j^{th} particle (in PSO)
$\boldsymbol{v}_k^{[n_j]}$	vector of n_j^{th} particle (in DiEv) at k^{th} iteration
$w \in \mathbb{R}^+$	inertia weight (in PSO)
w_{max} and w_{min}	maximum and minimum values of the weight parameter
$\boldsymbol{x}$	vector of design variables
$\boldsymbol{x}_k$	update for x at k^{th} iteration
$\hat{\boldsymbol{x}}_k$	trial state (in SA)
$\bar{\boldsymbol{x}}_k$	centroid of the first n best points of the simplex in NM method
$\boldsymbol{x}^{[j]}$	position vector of j^{th} particle (in PSO)
$\boldsymbol{x}^k$	final point of the k^{th} stage in rotating coordinates method of Rosenbrock
$\boldsymbol{x}_k^{CI}$	new variable obtained by contraction inside in NM method
$\boldsymbol{x}_k^{CO}$	new variable obtained by contraction outside in NM method
$\boldsymbol{x}_k^{E}$	new variable obtained by expansion in NM method
$\boldsymbol{x}_k^{R}$	new variable obtained by reflection in NM method
$\boldsymbol{x}_k^{S}$	update forming a new simplex in NM method
$x_{j,u}$ and $x_{j,l}$	upper and lower bounds of the design variable x_j
$y(t)$	vector of nodal displacements (translational and rotational)
z	observation data in Example 3.2

$\boldsymbol{Z}$	vector of RVs in Example 3.2
α	parameter in the generalized exponential *pdf* (Equation 3.13)
$\hat{\alpha}$	estimated parameter in Example 3.2
β	a parameter in rotating coordinates method of Rosenbrock
$- \in [\mathbf{0},\mathbf{1}]$,	a blending parameter in GA
ς_{rand}	randomly chosen integer in $[1, N]$ – in DiEv (Table 3.7)
Δ_k	trust region radius (step size)
∇	gradient (first-order derivative)
∇^2	second-order derivative matrix
λ	– parameter in the generalized exponential pdf (Equation 3.13) – Lagrangian multiplier (Equation 3.32)
$\hat{\lambda}$	estimated parameter in Example 3.2
μ_1 and μ_2	Lagrangian multipliers associated with equality constraints
μ_k	damping parameter (Equation 3.33)
ρ	mass density
$\psi(x)$	penalty function
$\boldsymbol{\theta}$	vector of unknown parameters (Example 3.2)
$\hat{\boldsymbol{\theta}}$	vector of estimated parameters (Example 3.2)
ω_1 and ω_2	natural frequencies

REFERENCES

Aarts, E. H. L. and J. H. Korst. 1989. *Simulated Annealing and Boltzmann's Machines: A Stochastic Approach to Combinatorial optimization and Neural Computing*. Wiley. NY.

Aarts, E. H. L. and P. van Laarhoven. 1985. Statistical cooling: a general approach to combinatorial optimization problems. *Philips Journal of Research* 40: 193–226.

Alba, E. and M. Tomassini. 2002. Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 6: 443–462.

Back, T. 1996. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press. Oxford.

Bartholomew-Biggs M. and S. C. Parkhurst. 2003. Global optimization approaches to an aircraft routing problem. *European Journal of Operational Research* 146(2): 417–431.

Begambre, O. and J. E. Laier. 2009. A hybrid particle swarm optimization – simplex algorithm (PSOS) for structural damage identification. *Advances in Engineering Software* 40(9): 883–891.

Bendsøe, M. P. and O. Sigmund. 2003. T*opology Optimization – Theory, Methods and Applications*. Springer Verlag. Heidelberg. Berlin.

Bland, J. A. 1994. A Tabu search approach to engineering optimisation. *Proceedings of the 9th International Conference on Application of Artificial Intelligence in Engineering*, pp. 423–430.

Brest, J., S. Greiner, B. Boskovic, M. Mernik, and V. Zumer. 2006. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, *IEEE Transactions on Evolutionary Computation* 10(6): 646–657.

Brezočnik, L., L. Fister, and V. Podgorelec. 2018. Swarm intelligence algorithms for feature selection: a review. *Applied Sciences* 8(9): 1521.

Byatt, D. 2000. *Convergent Variants of the Nelder-Mead Algorithm*. Master's Thesis. University of Canterbury. Christchurch, New Zealand.

Cerny, V. 1985. Thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications* 45(1): 45–51.

Chibante, R. 2010. *Simulated Annealing Theory with Applications*. Sciyo. Rijeka, Croatia.

Chong, E. K. P. and S. H. Zak. 2013. *An Introduction to Optimization*. 4th Ed. John and Wiley Sons.

Clerc, M. and J. Kennedy. 2002. The Particle Swarm: Explosion, Stability, and Convergence in a Multi-Dimensional Complex Space. *IEEE Transactions on Evolutionary Computation* 6: 58–73.

Clough, R. W. and J. Penzien. 1982. *Dynamics of Structures*. McGraw-Hill.

Conn, A. R, N. I. M. Gould, and P. L. Toint. 2000. *Trust-Region Methods*. SIAM. Philadelphia, USA.

Conn, A. R., K. Scheinberg, and L. N. Vicente. 2009. Introduction to derivative-free optimization. SIAM, Philadelphia, PA.

Conn, A. R., K. Scheinberg, and P. L. Toint. 1996. On the convergence of derivative-free methods for unconstrained optimization. In: M. D. Buhmann and A. Iserles (eds.), *Approximation Theory and Optimization*, Tribute to M. J. D. Powell, pp. 83–108. Cambridge University Press, Cambridge, UK.

Connor, A. M. and D. G. Tilley. 1998. A Tabu search method for the optimization of fluid power circuits. Journal of Systems and Control 212(5): 373–381.

Daniel, A. 2006. *Evolutionary Computation for Modelling and Optimization*. Springer. NY.

Deb, K. 1997. Genetic algorithm in search and optimization: The technique and applications. *Proceedings of the International Workshop on Soft Computing and Intelligent Systems*, pp. 58–87.

Deb, K. and S. Gulati. 2000. Design of truss-structures for minimum weight using genetic algorithms. *Finite Elements in Analysis and Design* 37(5): 447–465.

Dongarra, J. and F. Sullivan. 2000. Guest editors introduction to the top 10 algorithms. *Computing in Science and Engineering* 2(1): 22–23.

Doorly, D., J. Peiro, and J. Oesterla. 1996. Optimization of aerodynamic and coupled aerodynamic-structural design using parallel genetic algorithm. *Proceedings Of the 6th AIAA/NASA/ISSMD Symposium on Multi-disciplinary Analysis and Optimization*, pp. 401–409.

Dorigo, M., M. Birattari, and T. St ü tzle. 2004. Ant colony optimization: Artificial ants as a computational intelligence technique. Journal of *IEEE Computational Intelligence* 1(4): 28–39.

Dorigo, M., V. Maniezzo, A. Colorni. 1996. The ant system: optimization by a colony of cooperating agents. *IEEE Transactions on System, Man and, Cybernetics – Part B* 26(1): 29–41.

Duvigneau, R. and M. Visonneau. 2004. Hydrodynamic design using a derivative-free method. *Structural and Multidisciplinary Optimization* 28(2): 195–205.

Eby, D., R. Averill, E. Goodman, and W. Punch. 1999. The optimization of flywheels using an injection island genetic algorithm. In: Bentley, P. (ed.), *Evolutionary Design by Computers*, pp. 167–190. Morgan Kaufmann. San Francisco.

Engelbrecht, A. P. 2006. *Fundamentals of Computational Swarm Intelligence*. John Wiley and Sons. NY.

Fan, H. Y. and Lampinen, J. 2003. A trigonometric mutation operation to differential evolution. *Journal of Global Optimization* 27: 105–129.

Fisher, R. A. 1922. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London, Series A* 222: 309–368.

Fletcher, R. 1987. *Practical Methods of Optimization*. 2nd Ed. John Wiley & Sons. Ltd.

Fletcher, R. and C. M. Reeves. 1964. Function minimization by conjugate gradients. *Computer Journal* 7 : 149–154.

Glover, F. and G. A. Kochenberger. 2003. *Handbook of Metaheuristics*. Kluwer Academic Publishers.

Glover, F and M. Laguna. 1997. Tabu Search, Kluwer Academic Publishers.

Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st Ed. Addison-Wesley Publishing Company. Boston.

Goldberg, D. E., and M. P. Samtani. 1986. Engineering optimization via genetic algorithms. *Proceedings of the 9th Conference on Electronic Computing*, pp. 471–482. ASCE. NY.

Grimble, M. J. and M. A. Johnson. 1988. *Optimal Control and Stochastic Estimation*. Volumes 1 and 2. John Wiley and Sons. Chichester.

Guerlement, G., R. Targowski, W. Gutkowski, and J. Zawidzka. 2001. Discrete minimum weight design of steel structures using EC3 code. *Structural and Multidisciplinary Optimization* 22(4): 322–327.

Guirguis, D. and M. F. Aly. 2016. A derivative-free level-set method for topology optimization. *Finite Elements in Analysis and Design* 120: 41–56.

Gupta, R. D. and D. Kundu. 2003. Discriminating between the Weibull and the GE distributions. *Computational Statistics and Data Analysis* 43: 179–196.

Hager, W. W. 2001. Minimizing a quadratic over a sphere. *SIAM Journal of Optimization* 12: 188–208.

Hajek, B. 1988. Cooling schedules for optimal annealing. *Mathematics of Operations Research* 13: 311–329.

Hansen, N. 2007. *The CMA Evolution Strategy: A Tutorial*. Berlin: Springer-Verlag.

Hansen, N and A. Ostermeier. 1996. *Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation*, pp. 312–317. ICEC 96. IEEE Press

Haupt, R. L., 1995. An introduction to genetic algorithms for electromagnetics. *IEEE AP-S Magazine* 37(2): 7–15.

Haupt, R. L. and S. E. Haupt. 2004. *Practical Genetic Algorithms*. John Wiley & Sons, Inc., Hoboken, New Jersey.

He, Q. and L. Wang, 2007. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied Mathematics and Computation Journal* 186: 1407–1422.

Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Ann Arbor, MI.

Hooke, R. and T. A. Jeeves. 1961. Direct search solution of numerical and statistical problems. *Journal of the ACM* 8(2): 212–229.

Hu, X. and R. C. Eberhart. 2002. Solving constrained nonlinear optimization problems with particle swarm optimization. *Proceedings of the Sixth World Multiconference on Systemics, Cybernetics and Informatics*. SCI 2002. Orlando, USA.

Hu, M., T. Wu, and J. D. Weir. 2012. An adaptive particle swarm optimization with multiple adaptive method. *IEEE Transactions on Evolutionary Computation* 17: 1–15.

Hultman, Max. 2010. Weight optimization of steel trusses by a genetic algorithm – size, shape and topology optimization according to Eurocode. Report TVBK – 5176. Lund University. Lund. Sweden.

Jiao, B., Z. Lian, and X. Gu. 2008. A dynamic inertia weight particle swarm optimization algorithm. *Chaos, Solitons & Fractals* 37: 698–705.

Kang. F., Li. J, and Q. Xu. 2012. Damage detection based on improved particle swarm optimization using vibration data. *Applied Soft Computing* 12(8): 2329–2335.

Kelley. C. T. 1999. *Iterative Methods for Optimization*. SIAM, Philadelphia.

Kennedy J. 2006. Swarm intelligence. In: *Handbook of Nature-Inspired and Innovative Computing*, pp. 187–219. Springer.

Kennedy, J. and R. Eberhart, R. 1995. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks* 4: 1942–1948.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by simulated annealing. *Science* 220: 671–680.

Koza, J. R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

Krishnakumar, K. and D. E. Goldberg. 1992. Control system optimization using genetic algorithms. *Journal of Guidance, Control and Dynamics* 15(3): 735–740.

Lagarias, J. C,, J. A. Reeds, M. H. Wright, and P. E. Wright. 1998. Convergence properties of the Nelder–Mead simplex algorithm in low dimensions. *SIAM Journal on Optimization* 9: 112– 147.

Lehmann, E. L. and G. Casella. 1998. *Theory of Point Estimation*. Springer. NY.

Levenberg. K. 1944. A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics* 2: 164–168.

Lewis R. M. and V. J. Torczon. 1999. Pattern search algorithms for bound constrained minimization. *SIAM Journal on Optimization* 9: 1082–1099.

Lewis R. M. and V. J. Torczon. 2000. Pattern search algorithms for linearly constrained minimization. *SIAM Journal on Optimization* 10: 917–941.

Lewis R. M. and V. J. Torczon. 2002. A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Optimization* 12: 1075–1089.

Lewis, R. M., V. J. Torczon, and M. W. Trosset, 2000. Direct search methods: Then and now. *Journal of Computational and Applied Mathematics* 124: 191–207.

Lucidi, S. and M. Sciandrone. 2002. On the global convergence of derivative-free methods for unconstrained minimization. *SIAM Journal on Optimization* 13: 97–116.

Luh, G. C., C. Y. Lin, and Y. S. Lin. 2011. A binary particle swarm optimization for continuum structural topology optimization. *Applied Soft Computing* 11(2): 2833–2844.

Markowitz, H.M. 1952. Portfolio selection. *The Journal of Finance* 7(1); 77–91.

Mallipeddi, J. R. and P. N. Suganthan. 2008. Empirical study on the effect of population size on differential evolution algorithm. *Proceedings of the IEEE Congress on Evolutionary Computing*, pp. 3663–3670.

Marazzi, M. and J. Nocedal, 2002. Web trust region methods for derivative free optimization. *Mathematical Programming*, 91(2): 289–305.

Marquardt, D.W. 1963. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics* 11(2): 431–441.

Marsden, A. L, J. A. Feinstein, and C. A. Taylor. 2008. A computational framework for derivative-free optimization of cardiovascular geometries. *Computer Methods in Applied Mechanics and Engineering* 197: 1890–1905.

Marsden, A. L., M. Wang, J. E. Dennis Jr, and P. Moin. 2007. Trailing-edge noise reduction using derivative-free optimization and large-eddy simulation. *Journal of Fluid Mechanics* 572: 13–36.

McKinnon. K. I. M. 1998. Convergence of the Nelder-Mead simplex method to a non stationary point. *SIAM Journal on Optimization* 9: 148–158.

McLachlan, G. and D. Peel. 2000. Finite Mixture Models. Wiley Series in Probability and Statistics, John Wiley & Sons, Inc. NY.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. 1953. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics* 21(6): 1087–1092.

Michalewicz, Z. 1996. Genetic Algorithms + Data Structures = Evolution Programs. 3rd Ed. Springer Verlag.

Michalewicz, Z. and M. Schoenauer. 1996. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation* 4(1): 1–32.

Molga, M. and C. Smutnicki. 2005. Test Functions for Optimization Needs. Retrieved June 2013, from www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf.

More, J. J. and D. C. Sorensen. 1981. *Computing a Trust region Step*. Report ANL-81-83. Argonne National Laboratory. Argonne, Illinois.

Nelder J. A. and R. Mead. 1965. A simplex method for function minimization. *Computer Journal* 7: 308–313.

Newcomb, S. 1886. A generalized theory of the combination of observations so as to obtain the best result. *American Journal of Mathematics* 8: 343–366.

Niemi, J. B. 2009. Bayesian Analysis and Computational Methods for Dynamic Modelling. PhD thesis. Graduate School of Duke University. UK.

Nocedal, J. and S. J. Wright. 2006. *Numerical Optimization*. 2nd Ed. Springer-Verlag, NY.

Norets, A. and J. Pelenis. 2011. Posterior Consistency in Conditional Density Estimation by Covariate Dependent Mixtures. IHS Economics Series Working Paper 282.

Ogata, K. 1995. *Discrete-time Control Systems*. Prentice-Hall. Inc. 2nd Ed. NJ.

Ogata, K. 1997. *Modern Control Engineering*. 3rd Ed. Prentice-Hall, Inc. NJ.

Papoulis, A. 1991. *Probability, Random Variables and Stochastic Processes*. 3rd Ed. McGraw-Hill Inc., NY.

Pham, N. D. 2012. *Improved Nelder Mead's Simplex Method and applications*. PhD thesis. Auburn University.

Powell, M. J. D. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal* 7: 155–162.

Powell, M. J. D. 1970. A new algorithm for unconstrained optimization. In: J. B. Rosen, O. L. Mangasarian and K. Ritter (eds.) *Nonlinear Programming*, pp. 31–66. Academic Press. NY.

Price, K. V., R. M. Storn. and J. A. Lampinen. 2005. *Differential Evolution. A Practical Approach to Global Optimization*. Springer-Verlag. Berlin, Heidelberg.

Quandt, R. E. and J. B. Ramsey. 1978. Estimating Mixtures of Normal Distributions and Switching Regressions. *Journal of the American Statistics Association* 73(364).

Ravindran, A., K. M. Ragsdell, and G. V. Reklaitis. 2006. Engineering Optimization, Methods and Applications. 2nd Ed. John Wiley & Sons. UK.

Rios, L. M. and N. V. Sahinidis. 2013. Derivative-free optimization: A review of algorithms and comparison of software implementations. *Journal of Global Optimization* 56(3): 1247–1293.

Rosenbrock, H. H. 1960. An automatic method for finding the greatest or least value of a function. *Computer Journal* 3: 175–184.

Roy, D. and G.V. Rao. 2017. *Stochastic Dynamics, Filtering and Optimization*. Cambridge University Press. UK.

Schmidt, P. 1982. An improved version of the Quandt-Ramsey MGF estimator for mixtures of normal distributions and switching regressions. *Econometrica* 50(2).

Shi Y and R. Eberhart. 1998. A modified particle swarm optimizer. In: *The 1998 IEEE International Conference on IEEE World Congress on Computational Intelligence. Evolutionary Computation Proceedings*, pp. 69–73.

Silva Neto, A. J. and M. N. Özişik. 1994. The estimation of space and time dependent strength of a volumetric heat source in a one-dimensional plate. *International Journal of Heat and Mass Transfer* 37(6): 909–915.

Slowik, A. and H. Kwasnicka. 2018. Nature inspired methods and their industry applications—Swarm intelligence algorithms. *IEEE Transactions on Industrial Informatics* 14(3): 1004–1015.

Smith, J. and T. Fogarty. 1997. Operator and parameter adaptation in genetic algorithms. *Soft Computing* 1(2): 81–87.

Sorensen, D. C. 1982. Newton's method with a model trust region modification. *SIAM Journal on Numerical Analysis* 19(2): 409–426.

Sotiropoulos, D. G., E. C. Stavropoulos, and M. N. Vrahatis. 1997. A new hybrid genetic algorithm for global optimization. *Nonlinear Analysis, Theory, Methods & Applications* 30(7): 4529–4538.

Souza, F.L., F. J. C. P. Soeiro, A. J. Silva Neto, F. M. and Ramos. 2007. Application of the generalized external optimization algorithm to an inverse radiative transfer problem. *Inverse Problems in Science and Engineering* 15(7): 699–714.

Srinivas, M. and L. Patnaik. 1994. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Trans. on Systems. Man and Cybernetics* 24(4): 656–667.

Steihaug, T. 1983. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis* 20(3): 626–637.

Storn, R. and K. Price. 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11: 341–359.

Toint. Ph. L. 1988. Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space. *IMA Journal of Numerical Analysis* 8(2): 231–252.

Torczon. V. J. 1991. On the convergence of multidirectional search algorithms. *SIAM Journal on Optimization* 1: 123–145.

Torczon. V. J. 1997 On the convergence of pattern search algorithms. *SIAM Journal on Optimization* 7: 1–25.

Tseng, P., 1999. Fortified-descent simplicial search method: A general approach. *SIAM Journal on Optimization* 10: 269–288.

Van den Bergh, F. and A. P. Engelbrecht. 2004. A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation* 8: 225–239.

Van Laarhoven, P. J. M. and E. H. L. Aarts 1987. *Simulated Annealing: Theory and Applications*. Springer Science & Business Media.

Venter. G. and J. Sobieszczanski-Sobieski. 2004. Multidisciplinary optimization of a transport aircraft wing using particle swarm optimization. *Structural and Multidisciplinary Optimization* 26 (1–2): 121–131.

Walters, F. H., L. R. Parker, S. L. Morgan, and S. N. Deming. 1991. *Sequential Simplex Optimization*. CRC Press, Boca Raton, FL.

Wang, Y., Z. X. Cai, and Q. F. Zhang. 2011. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation* 15(1): 55–66.

Wong, K. L. and A. G. Constantinides. 1998. Speculative parallel simulated annealing with acceptance prediction. *Electronics Letters* 34(3): 312–313.

Wright, M. H. 1996. Direct search methods: Once scorned, now respectable. In: Numerical Analysis. *Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis*. D. F. Griffiths and G. A. Watson. (eds.), pp. 191–208. Addison Wesley Longman. Harlow, UK.

Zhao, X., C. Wang, J. Su, and J. Wang. 2019. Research and application based on the swarm intelligence algorithm and artificial intelligence for wind farm decision system. *Renewable Energy* 134: 681–697.

FURTHER READINGS

Aarts, E. H. L. and Korst, J. 1989. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. John Wiley & Sons, NY.

Audet, C. and Dennis Jr., J. E. 2002. Analysis of generalized pattern searches. *SIAM Journal on Optimization* 13(3): 889–903.

Durand, N. and Alliot, J. 1999. A combined Nelder-Mead simplex and genetic algorithm. In: *Proceedings of the Genetic and Evolutionary Computation Conference* 99: 1–7.

Harris, J. W. and Stocker, H., 1998. Maximum likelihood method. In: *Handbook of Mathematics and Computational Science*, p. 824. New York: Springer-Verlag.

Kolda, T. G., Lewis, R. M. and V. Torczon. 2003. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review* 45(3): 385–482.

Marsden, A. L., Feinstein, J. A. and Taylor, C. A. 2008. A computational framework for derivative-free optimization of cardiovascular geometries. *Computational Methods in Applied Mechanics and Engineering* 197: 1890–1905.

Mckinnon, K. I. M., 1998. Convergence of the Nelder-Mead simplex method to a nonstationary point. *SIAM Journal on Optimization* 99(1): 148–158.

Price, C. J., Coope, I. D. and Byatt, D. 2002. A convergent variant of the Nelder-Mead algorithm. *Journal of Optimization Theory and Applications* 11(3): 5–19.

4 Elements of Riemannian Differential Geometry and Geometric Methods of Optimization

4.1 INTRODUCTION

Optimization techniques described in Chapters 1 to 3 are classical in the sense that they are based on Euclidean geometry. The design variable space is Euclidean $\mathbb{R}^n$ which is a vector space endowed with the familiar distance metric

$$d(\boldsymbol{x},\boldsymbol{y}) = \sqrt{(\boldsymbol{x}-\boldsymbol{y}).(\boldsymbol{x}-\boldsymbol{y})} \tag{4.1}$$

where *dot* within the square root sign on the right hand side (RHS) of Equation (4.1) signifies the vector dot product. Using the standard tools of differential calculus in $\mathbb{R}^n$, we could adopt, for instance, a line search technique to arrive at the optimum. The line direction would be guided by the classical definition of directional derivative. As an example, see the result in Figures 4.1a–b on the Rosenbrock function by the (classical) CG method (Section 2.2.2, Chapter 2).

An alternative to this classical approach based on line search may be to look for an optimum by a curved search also known as geodesic search on the curved hypersurface (also called a manifold) S in $\mathbb{R}^n$. The latter is defined by a given cost function and/or a set of constraints (in general, we will denote a manifold by the letter *M*). The idea of a curved search on S has origins in the method of geodesic descent mooted by Luenberger (1972). A geodesic is defined as a smooth curve $\gamma(t)$ on *S*, $0 \le t \le T$, starting at $a = \gamma(0)$ and terminating at $b = \gamma(T)$ that minimizes the function $\int_0^T \|\dot{\gamma}(t)\| dt$ among all other such curves between a and b on *S*. The overdot on γ denotes the first-order derivative with respect to the parameter *t*. $\gamma(t)$ may be represented by the coordinate functions $\{x^1(t), x^2(t), .., x^n(t)\}$. In optimization algorithms, the parameter t may be thought of as an arc-length or a pseudo time variable over which iterations are defined. In this chapter and Chapter 5, we are interested in learning the basic principles that govern such a geometric search. Luenberger (1972) has mainly restricted the idea of geodesic search to theoretical considerations alone in that the construction has been used to prove the convergence

DOI: 10.1201/9781003414063-4

properties of the gradient projection method of Rosen (1960,1961) (see Section 2.8, Chapter 2). Figure 4.2 outlines the scheme involved in Rosen's method. Leaving the formal definition of a manifold to the next section in this chapter, we may at this stage view a manifold as a set of feasible points that form a smooth surface.* The constraining surface in Figure 4.1a is in fact a two-dimensional manifold embedded in $\mathbb{R}^3$ and represents an injective map $F:\mathbb{R}^2 \to \mathbb{R}^3$ with $F(x,y) = (x,y,f(x,y))$ where f is the cost function.

The computational exercise of actually finding a geodesic on a manifold is not easy (Smith 1993, Boumal 2014) especially when the manifold dimension is high. The variants (Botsaris 1981a,b) of Luenberger's idealized version of geodesic search mainly focussed on how to avoid the complexity of finding the geodesic path and yet generate a new update close to S in the last step of the gradient projection algorithm (Figure 4.2).

However, with literature (Boothby 1975, Lang 1995, Hsu 2002, Lee 2003, Tu 2011) significantly increasing on the theory of manifolds in the latter half of the last century, renewed research efforts are under way on developing optimization methods using the intrinsic structure of manifolds. The structure mainly includes the differential geometric aspects of manifolds, such as tangent spaces, metric, geodesic, covariant derivative, vector transport and incompatibility tensors such as curvature. The word 'intrinsic' is sometimes used when these structures are described without explicitly using the coordinates of the ambient (Euclidean) space in which the manifold may be embedded. In Section 4.2, we describe these geometric aspects before moving on to the geometric methods of optimization based on manifolds. Section 4.3 presents some of these geometric methods which are Riemannian analogues of a few of the classical optimization methods discussed in Chapter 2. One finds applications of these algorithms in myriad fields (Absil *et al*. 2007a,b, Baker 2008) including continuum mechanics (Aubram 2009), machine learning (Lin and Jha 2008), filtering (Hauberg et al. 2013), computer vision (Turaga et al, 2008) and statistical estimation (Hosseini and Sra 2015, Hajri et al. 2017). The problem of statistical estimation which has been dealt with in Chapter 3 by classical methods of optimization is reconsidered in this chapter (Section 4.4) and solved by the Riemannian version.

Against the category of problems so far discussed and posed in a deterministic setting, one often encounters many problems which are stochastic (either inherently so or posed stochastically to reap certain advantages). The classical evolutionary optimization methods described in Chapter 3 (Section 3.4) indeed belong to this

---

* smooth surface
A smooth surface is a collection of points, each with a tangent plane that continuously varies from point to point. Suppose that a surface S is parametrized via the map $\pi: U \subset \mathbb{R}^2 \to S \subset \mathbb{R}^3$. With (u,v) being the coordinates of a point in U, S is said to be smooth if the components $\pi_1(u,v), \pi_2(u,v)$ and $\pi_3(u,v)$ have continuous partial derivatives with respect to u and v up to all orders.

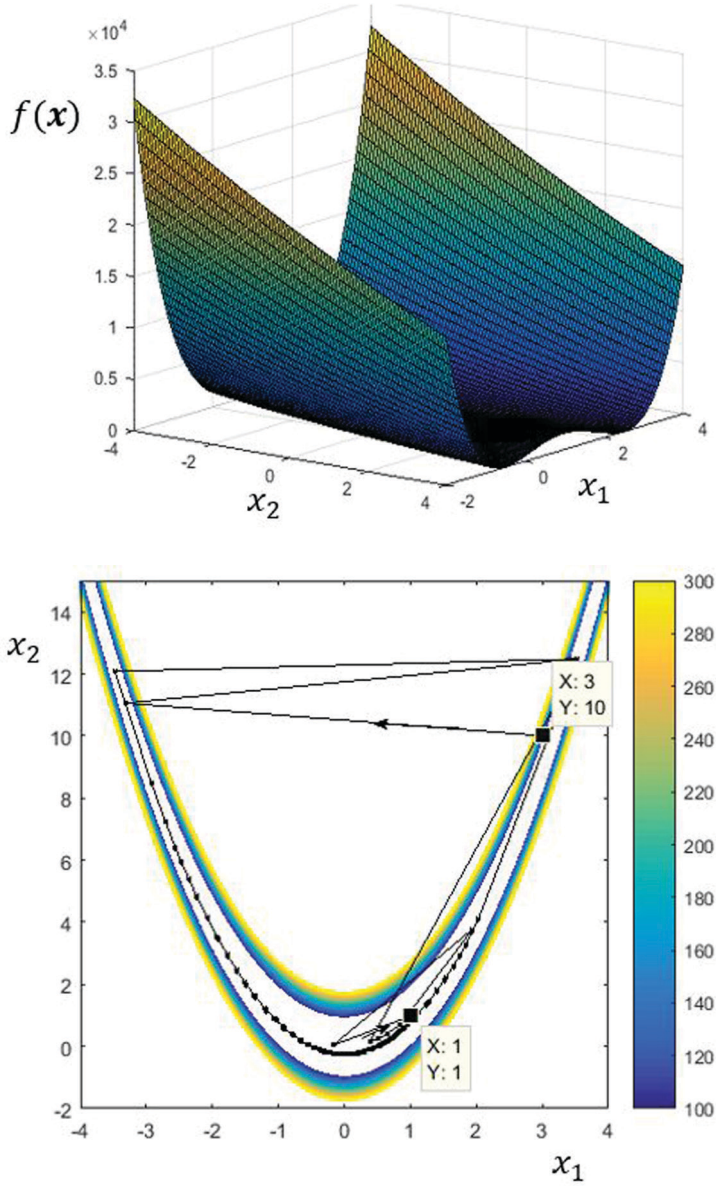


FIGURE 4.1a–b Rosenbrock function $f(\boldsymbol{x}) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2$ with $x = \left(x_1, x_2\right)$; contour plot in $\mathbb{R}^3$. Optimization in Euclidean space of Rosenbrock function $f(\boldsymbol{x}) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2$ with $\boldsymbol{x} = \left(x_1, x_2\right)$; projection of contour on to $\mathbb{R}^2$ and route to optimum – line with dots – (minimum) point $\boldsymbol{x}^* = (1,1)$ by line search (classical CG method).

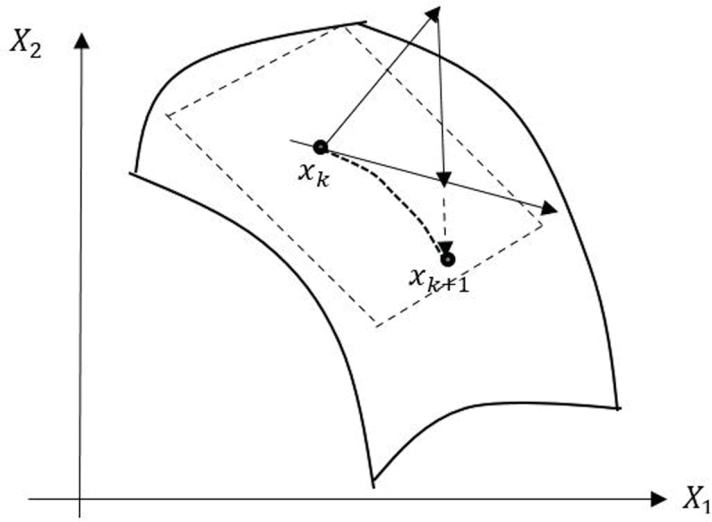


FIGURE 4.2 Gradient projection method of Rosen (1960, 1961).

category of methods based on stochastic search with metaheuristic origin (Holland 1975). While a stochastic route often facilitates an effective search, the posing of the original problem could itself be deterministic, wherein the aim would merely be to arrive at the design point that is the global extremum of an (or a collection of) objective functional(s), possibly subject to a set of prescribed constraints. With the exception of SA, these so-called stochastic search methods are mostly founded on elementary concepts of probability theory and Monte Carlo simulation (Appendix 3), and offer no guarantee of attaining the global optimum. The methodology in SA subsumes a deeper understanding of the MCMC methods (Appendix 3) that depend on the theory of Markov chains. In a more general context, MCMC has been employed for developing some of the most effective stochastic optimization algorithms (Durmus *et al.* 2019, Mamajiwala and Roy 2022). This is indeed facilitated by the strong analogy between statistical sampling by MCMC algorithms and optimization methods (Dalalyan 2017a). Obviously, the exposition on these methods requires a prior understanding of stochastic processes, stochastic calculus and solutions of SDEs (Roy and Rao 2017). A short discussion on these aspects is provided in Appendix 4.

Wind/seismic events are some of the familiar examples of stochastic processes in engineering applications; see Figure 4.3 for a few typical wind velocity profiles. A stochastic process is, roughly speaking, a random variable that, being parameterized in time, can evolve. Basic concepts in probability theory – random variables, probability distributions and expectations (statistical moments) – and their exploitations in computer simulations are outlined in Appendix 1.

Despite the involved nature of the concepts to be discussed in this chapter, we have tried to present the basics by shunning rigour in favour of clarity wherever possible.

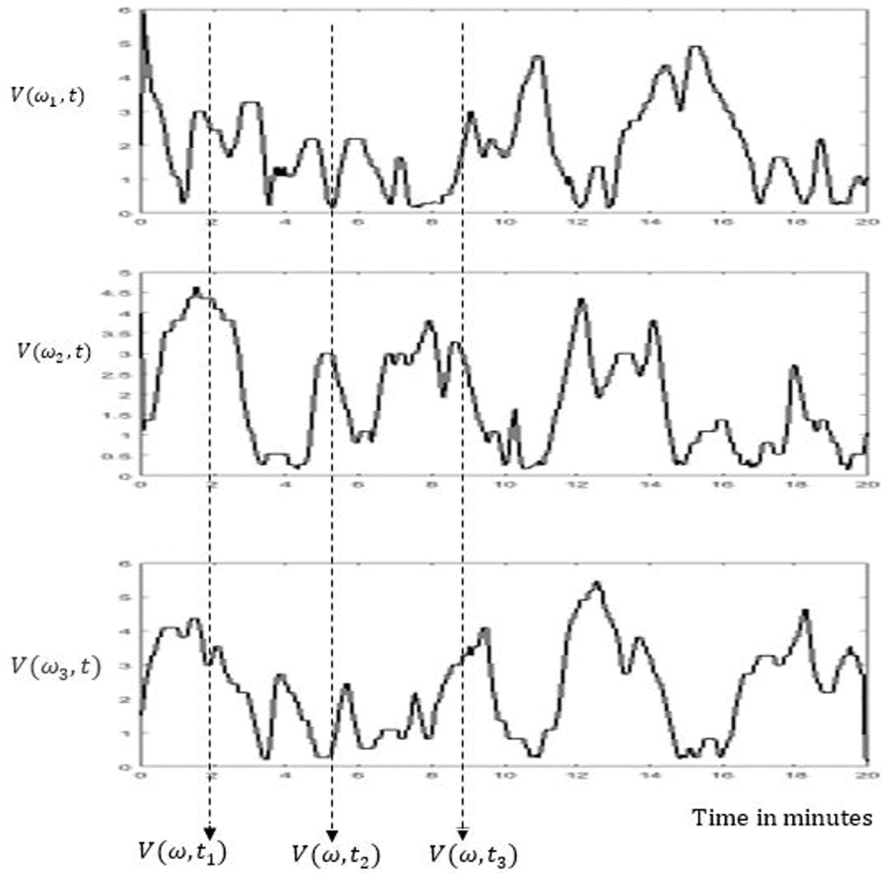


FIGURE 4.3 Three typical wind velocity profiles (refers to no specific real data).

We have done so with the hope that a beginner will get an intuitive understanding of the ideas with relative ease. In view of the importance of the Langevin SDE in the development of optimization methods, we give in Section 4.5 more details on this SDE. In fact, this SDE models the motion of a massive particle moving in a viscous fluid under a white noise process and is widely used in stochastic modelling. The equation is often considered a physical basis for the theory of Brownian motion in view of the early work by Einstein (1905). To motivate further, we may as well provide here a few more details about this SDE. Also, the Langevin equation occupies a prime place in describing the geometric methods of optimization in the present chapter and the next one. The equation in terms of the position $x(t)$ and velocity $v(t)$ of a particle is (Zwanzig 2001):

$$\dot{x}(t) = v(t)$$

$$\dot{v}(t) = -\frac{c}{m}v(t) + \frac{1}{m}w(t) \tag{4.2a,b}$$

m is the mass of the particle and c the damping coefficient due to friction during bombardments of the particle by neighbouring fluid molecules. c is related to the viscosity of the fluid. $w(t)$ is the randomly fluctuating force which is due to the interaction of the particles with the surrounding medium (heat bath). In other words, these random fluctuations are thermally activated. $w(t)$ is thus modelled as a white noise – with zero mean, i.e. $E\left[w(t)\right] = 0$ and varying so rapidly that the correlation between two distinct time instants t and t' (no matter how close these two times are) is zero, i.e.

$$E\left[w(t)w(t')\right] = 2\sigma\delta(t - t') \tag{4.3}$$

where $\delta(.)$ is the Dirac delta function. 2σ is the strength of the noise. In the absence of the noise term in Equation (4.2b), it is obvious that the velocity of the particle decays to zero as $t \to \infty$. However, early investigations by the biologist Robert Brown (1827), and later on by Einstein (1905), brought to the fore that the randomly fluctuating force which is invariably present depends on both friction and temperature. In fact, from the solution to Equation (4.2), one may show (see Zwanzig 2001 for details) that the mean squared velocity approaches the equilibrium value $K_B T/m$ where K_B is the Boltzmann constant and T is the absolute temperature. Thus, for time large enough, the solution reaches a steady state, specifically a thermal equilibrium where the frictional force drives the system towards a "dead" state even as the random force (noise) keeps it "alive". This interplay of two conflicting aspects reaching a thermal equilibrium has led Einstein to propose the earliest version of what is today called the fluctuation-dissipation theorem. In the formalism by Einstein (1905), the equilibrium solution is directly derived in terms of a PDE known as the Fokker-Planck equation whose solution gives the underlying *pdf*.

One perceives that the solution to the Langevin SDE reaching a stationary (that is to say, time independent) *pdf* may be utilized as an MCMC sampler without the need for a proposal *pdf*. See Appendix 3 for a discussion on MCMC methods. This is indeed exploited in developing sampling algorithms (Welling and Teh 2011, Martin *et al.* 2012, Wibisono 2018, Durmus *et al.* 2019) useful in statistical estimation and system identification problems. To highlight the significance of the analogy between sampling and optimization in the development of optimization methods, Section 4.5 is devoted to a discussion particularly on Langevin dynamics. Taking cue from this analogy, we proceed to discuss in Section 4.6 analogous optimization strategies for function optimization posed in the stochastic setting, by first presenting a classical method of this genre followed by a geometric version.

4.2 MANIFOLDS, LOCAL EUCLIDEAN PROPERTY AND CHARTS

Curves and surfaces we are generally familiar with are manifolds and non-Euclidean. However, a manifold M could be locally Euclidean. For instance, a small neighbourhood of a point p on a curve (in R^2 or R^3) is almost a straight line (see Figure 4.4).

The surface of the earth is a manifold embedded in R^3 and it is known to have a representation in R^2 through charts and an atlas. The reader may have heard about a chart and an atlas. For present purposes, however, it is essential that these concepts are learnt with certain clarity. At any location on the plains, the surface of the earth seems locally flat, just like a two-dimensional plane. Standing on an open meadow, it is difficult to notice the curvature of the earth with bare eyes. This locally Euclidean property means that there exists an open neighbourhood $U \subset M$ containing the point p and a homeomorphism $\varphi : U \to U'$ where U' is an open set in R^n, for some positive integer n denoting the manifold dimension. Homeomorphism means that φ is a bijective map (i.e. a one-to-one-map) with φ and φ^{-1} continuous (but not necessarily differentiable). The pair (U, φ) is known as a coordinate chart or simply a chart. Figure 4.5 shows the mapping φ for a sphere (also called S^2; we use S^1 for a circle).

For mapping the earth's surface, as is known, we must have a number of charts, i.e. more than one of them. A collection $\mathcal{A}$ of such charts on M is called an atlas where any two charts smoothly overlap (i.e. they are compatible) and open sets U cover M, i.e., for every $p \in M$ there is a coordinate chart $(U, \varphi) \in \mathcal{A}$ with $\varphi(p) \in U'$. Compatibility of two charts (U, φ) and (V, Ψ) is defined as follows. Consider the subsets $\varphi(U \cap V)$ and $\Psi(U \cap V)$ (see Figure 4.6).

If these subsets are open and the transition map $\Psi \circ \varphi^{-1} : \varphi(U \cap V) \to \Psi(U \cap V)$ is a diffeomorphism, then the two charts (U, φ) and (V, Ψ) are compatible. A map F is a diffeomorphism if it is a homeomorphism and is also C^∞ differentiable (smooth). If $U \cap V = \varnothing$, the charts are trivially compatible. Notice that the reverse

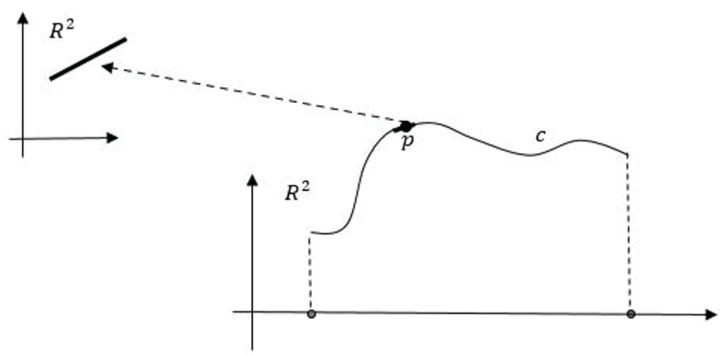


FIGURE 4.4a Local Euclidean property; curve in R^2.

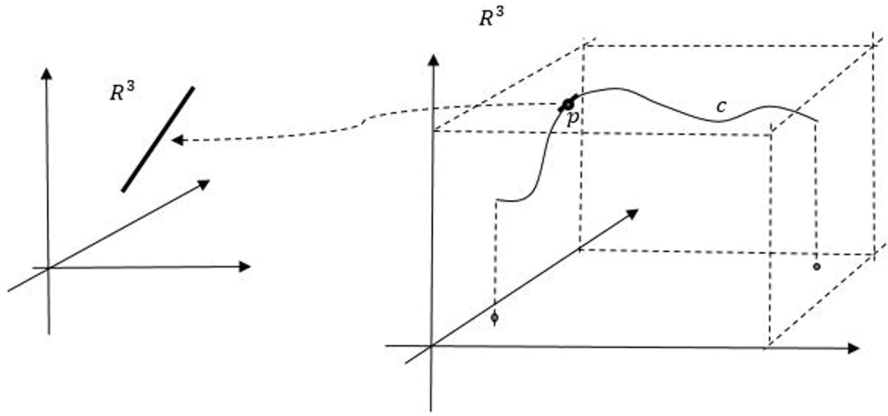


FIGURE 4.4b Local Euclidean property; curve in R^3.

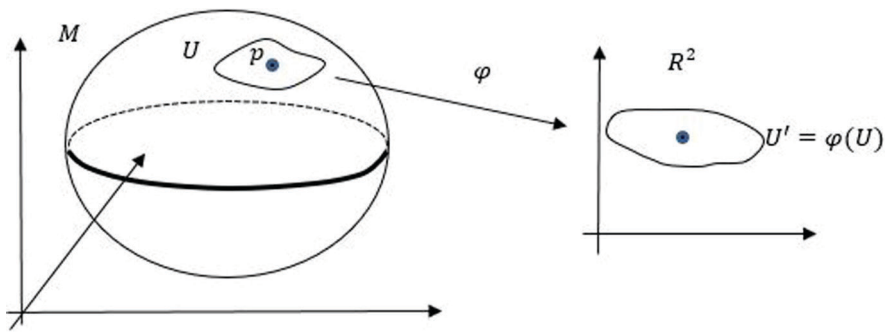


FIGURE 4.5 Manifold M and coordinate chart (U, φ) where $U \subset M$.

transition map $\varphi \circ \Psi^{-1}$ being the inverse of $\Psi \circ \varphi^{-1}$ is also a diffeomorphism. We may now define a smooth (differentiable) manifold as a pair consisting of a set M and a maximal atlas $\mathcal{A}$ on M. A maximal atlas is the one in which every chart is smoothly compatible with each of its members. The compatibility condition implies that whenever a pair of charts overlap, the charts exhibit approximately the same view of the manifold in those overlapping parts.

The chart (U, φ) assigns a coordinate system to the neighbourhood of any point $p \in M$ and the point p acquires the coordinates $\varphi^1(p)$, $\varphi^2(p), \ldots, \varphi^n(p)$ in $U' \subset R^n$. Thus, via these mappings, the manifold M attains a differentiable structure enabling differentiation and integration operations on M.

Implicit function theorem

Representation of manifolds by charts is indeed facilitated by the implicit function theorem (Lee 2003). The implication of the theorem is as follows. Consider a vector

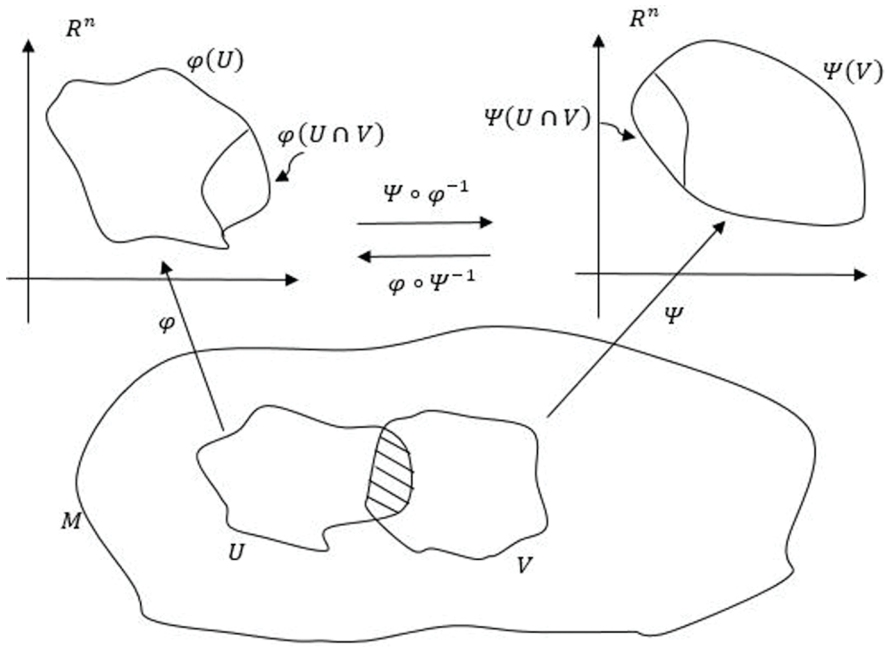


FIGURE 4.6 Manifold M and compatibility of two coordinate maps φ, Ψ via transition maps $\varphi \circ \Psi^{-1}$ and $\Psi \circ \varphi^{-1}$.

function $\boldsymbol{F}(\boldsymbol{x},\boldsymbol{y}): R^{m+n} \to R^n$ where $\boldsymbol{x} \in R^m$ and $\boldsymbol{y} \in R^n$. Let $\boldsymbol{F}$ be differentiable with the Jacobian $d\boldsymbol{F} \in R^n \times R^{m+n}$. If the partitioned matrix $[d\boldsymbol{F}]_y$ representing the derivatives of $\boldsymbol{F}$ with respect to $\boldsymbol{y}$ has full rank equal to n at a point $(\boldsymbol{x},\boldsymbol{y})$ with $\boldsymbol{F}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{0} \in R^n$, then there exists an open neighbourhood $U' \subseteq R^m$ of $\boldsymbol{x}$ and a differentiable map $\boldsymbol{\mathcal{G}}: U' \to R^n$ such that $\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{\mathcal{G}}(\boldsymbol{x})) = \boldsymbol{0}$ for $\forall \boldsymbol{x} \in R^m$.

To make the implication of the theorem clear, we invoke the example of the unit sphere $S^2 = \{x, y, z \mid x^2 + y^2 + z^2 - 1 = 0\}$. We let $\boldsymbol{x} = (x, y)$ and $\boldsymbol{y} = z$, i.e. $m = 2$ and $n = 1$. Further, $F(\boldsymbol{x},\boldsymbol{y}) = F(x, y, z) = x^2 + y^2 + z^2 - 1$ and

$$dF = (2x, 2y, 2z), [dF]_z = 2z \tag{4.4}$$

$[dF]_z$ is of rank equal to $n = 1$ for all $(x, y, z) \in F^{-1}(0) \Rightarrow z = \mathcal{G}(x, y) = \pm\sqrt{1 - x^2 - y^2}$, i.e., $F(x, y, \mathcal{G}(x, y)) = 0$ for $\forall x, y \in R^2$. This leads to the open neighbourhood U', the $x - y$ plane which is in R^2 ($m = 2$) and which is the dimension of the manifold S^2. The dimension of the chart is the same as that of the manifold. Note that the manifold

S^2 may be represented by just two coordinate charts, one with $z = +\sqrt{1-x^2-y^2}$ for the upper half of the sphere and the other with $z = -\sqrt{1-x^2-y^2}$ for the lower half.

Before moving on to the next subsection to describe tangent vectors and the tangent space on a manifold, a few additional remarks on manifolds will be in order. (a) If the locally Euclidean property holds, a manifold M is also a topological space whose elements are open sets. M is generally referred to as a topological manifold. (b) If the smoothness property of a manifold is relaxed in that if, instead of the C^∞ differentiability, each transition map is of the class C^k, then M is a C^k manifold. However, in the rest of our discussion on manifolds, smoothness is assumed to be C^∞. (c) If M is a smooth manifold, a function $F : M \to R$ is called smooth if for every chart (U, φ) on M, $F \circ \varphi^{-1}$ is smooth on $\varphi(U) \subseteq R^n$. A set of such smooth functions on a manifold M is denoted by $C^\infty(M)$. We also define $C^\infty(p)$ with $p \in M$ as a set of smooth functions $F : U \to R$ where $U \subset M$ is an open set containing p. (d) In a similar manner, one may define smoothness of functions between two smooth manifolds M and N (Lee 2003).

4.2.1 Tangent Vectors and Tangent Space on Manifolds

In a Euclidean space, say R^n equipped with a vector space (inner product) structure, definition of a tangent vector is straightforward. That is, given a smooth curve $\gamma(t) : R \to R^n$, the tangent vector at a point $p = \gamma(t_0)$ on the curve is $\left.\frac{d\gamma}{dt}\right|_{t_0}$ or simply $\dot{\gamma}(t_0)$. In terms of the coordinate function, $\dot{\gamma}(t_0) = \left(\dot{x}^1(t_0), \dot{x}^2(t_0), \ldots, \dot{x}^n(t_0)\right)$. With $\dot{\gamma}(t_0)$ denoted by v, we often have to make use of a tangent vector to get the directional derivative $D_v(f)$ of a smooth function $f : C^\infty(R^n) \to R$. The derivative which stands for the instantaneous change of f along $\gamma(t)$ at the point p is:

$$D_v\left(f(p)\right) = \left.\frac{df(p+tv)}{dt}\right|_{t=0} = \nabla f_p . v \tag{4.5}$$

∇f_p is called the gradient of the function f at p. Note that the symbols $f(p)$ and f_p are being used synonymously. If v^i are the components of v along the standard orthonormal bases $e_i = \frac{\partial}{\partial x^i}$, $i = 1, 2, .., n$, then one has:

$$D_v\left(f(p)\right) = v^i \frac{\partial f_p}{\partial x^i} \tag{4.6}$$

Einstein convention† is implied in Equation (4.6).

If we define a linear map $\chi : C^{\infty}\left(R^n\right) \to R$ satisfying the product rule (the Lebnitz property):

$$\chi\left(f\left(p\right)h\left(p\right)\right) = f_p\chi\left(h_p\right) + h_p\chi\left(f_p\right) \tag{4.7}$$

where $f,\ h \in C^{\infty}\left(R^n\right)$, then D_v is one such map. The linear map χ is known as a "derivation" at the point p. The set of all derivations of $C^{\infty}\left(R^n\right)$ at p or, equivalently the set of all vectors at point p forms the tangent space $T_p\left(R^n\right)$. It is a vector space under the usual vector addition and scalar multiplication. Similar definition of a tangent vector is possible for a general manifold M. However, we need to utilize the local Euclidean property of manifolds.

Consider a smooth M and a function $f : C^{\infty}\left(M\right) \to R$. Let $\gamma\left(t\right)$ be a smooth map $\gamma : \left(-\varepsilon, +\varepsilon\right) \to R^n$ as shown in Figure 4.7 with $p = \gamma\left(0\right) \in U \subset M$.

Now, $f \circ \gamma : R \to R$ is a real valued function defined on $t \in \left(-\varepsilon, +\varepsilon\right)$. In the absence of a vector space structure on a manifold, it is not possible to differentiate a path as in the Euclidean case. However, we take recourse to the notion of directional derivative to define a tangent vector on M.

Accordingly, $\left(\dot{\overline{f \circ \gamma}}\right)\left(t\right)$ may be interpreted as the manifold version of the directional derivative of f along the curve $\gamma\left(t\right)$ at any t. By the local Euclidean property of a manifold, we express this derivative using the coordinate representation through a chart.

Let $\left(U, \varphi\right)$ be a coordinate chart such that $\gamma\left(-\varepsilon, +\varepsilon\right) \subset U$. Then we have:

$$\left(f \circ \gamma\right) = \left(f \circ \varphi^{-1}\right) \lozenge \left(\varphi \circ \gamma\right) \tag{4.8}$$

The evaluation of the directional derivative follows by the familiar chain rule:

$$\dot{\overline{\left(f \circ \gamma\right)}}\left(t\right) = \frac{dF}{dX} \cdot \frac{dX}{dt}\bigg|_t \tag{4.9}$$

---

† Einstein convention implies an implicit sum over indices appearing twice in lower and/or upper positions in an expression. For example, the RHS of Equation (4.6) implies a summation over $i = 1, 2, \ldots, n$ since it appears in the upper position in v^i and lower position in $\frac{\partial f_p}{\partial x^i} = \partial_i f_p$, i.e:

$$D_v\left(f\left(p\right)\right) = \sum_{i=1}^{n} v^i \partial_i f_p \tag{i}$$

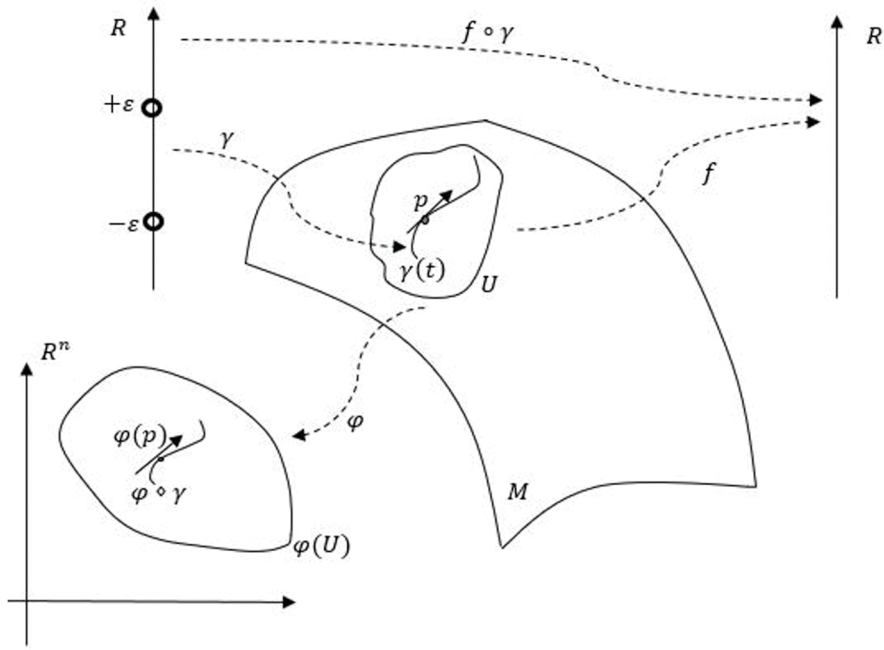


FIGURE 4.7 Manifold M; definition of a tangent vector using a coordinate chart (U,φ).

where $F = f \circ \varphi^{-1}$ and $X = \varphi \circ \gamma$. With $\gamma(t) = \left(x^1(t), x^2(t), .., x^n(t)\right)$ representing any point $x(t) \in U \subset M$, X represents the local coordinates $\left(\varphi\left(x^1(t)\right), \varphi\left(x^2(t)\right), .., \varphi\left(x^n(t)\right)\right)$. $\left.\frac{dX}{dt}\right|_t = \left(\overrightarrow{\varphi \circ \gamma}\right)(t) = v$, say, denotes the tangent vector to the curve $\gamma(t)$ on the manifold M (via the coordinate representation by the chart (U,φ)) with the vector components:

$$v^i(p) = \left.\frac{dX^i}{dt}\right|_t, \; i = 1,2,..,n \tag{4.10}$$

Equation (4.9) indicates the dependence of the directional derivative on the curve γ. For the directional derivative to be the same, curves passing through p need to have the same components $v^i(p), i = 1,2,\ldots,n.$ Such curves form an equivalence class of curves. Any two curves $\gamma_1 : \left(-\varepsilon_1, +\varepsilon_1\right) \to R^n$ and $\gamma_2 : \left(-\varepsilon_2, +\varepsilon_2\right) \to R^n$ of this class are tangent to each other at p. In other words, such an equivalence class of curves gives the same directional derivative at $p \in U \subset M$, that is:

$$\dot{\overline{(f\circ\gamma)}}(t)=\dot{\overline{(\varphi\circ\gamma_2)}}(t) \tag{4.11}$$

Within a particular chart (U,φ), the n components $v^i(p)$ (Equation 4.10) uniquely represent a tangent vector at p. Thus, for a general manifold, the tangent space at p denoted by T_pM (Figure 4.8) may be defined as a set of directional derivatives using an equivalence class of curves.

It also follows from Equations (4.9–4.10) that the differential operator $v^i\dfrac{\partial}{dX^i}$ denoted, say, by $D_{M,v}$ belongs to the linear map $\chi^M: C^\infty(M)\to R$ satisfying the Leibniz property (as in the Euclidean case):

$$\chi^M\left(f(p)h(p)\right)=f_p\chi^M\left(h_p\right)+h_p\chi^M\left(f_p\right) \tag{4.12}$$

The linear map is called a derivation (or simply a vector) at p on the manifold M. We note that v is a tangent vector and an element of T_pM, if it defines a derivation $\chi^M(f)$ on functions $f: C^\infty(M)\to R$. The set of all derivations at p constitutes a tangent space T_pM which is indeed a vector space under the operations:

$$\left(\chi^M+\mathcal{X}^M\right)f=\chi^M f+\mathcal{X}^M f \tag{4.13}$$

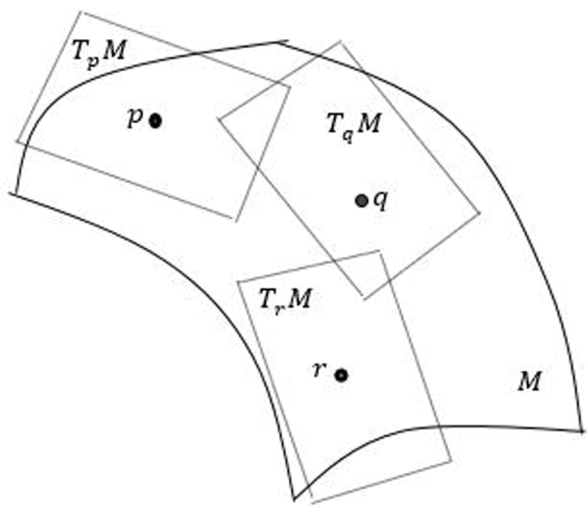


FIGURE 4.8 Manifold M; tangent spaces at points p, q and r.

$$\left(c\chi^M\right)f = c\left(\chi^M f\right) \tag{4.14}$$

where χ and $\mathcal{X}$ are derivations and c is a scalar constant. By the chain rule involved in Equation (4.9), the definition of a tangent vector through a derivation is independent of the chart (coordinates) chosen.

Tangent bundle

The set $\left\{T_pM \mid p \in M\right\}$ is called a tangent bundle *TM*, i.e.:

$$TM = \left\{(p,v): p \in M, v \in T_pM\right\} \tag{4.15}$$

The tangent bundle has the structure of a differentiable manifold in itself. If the tangent vector v is of dimension n, TM is a $2n$-dimensional manifold with M as the base space.

Cotangent vectors and cotangent space

A Euclidean vector space V has a dual vector space V^* defined via a linear functional $\mathrm{F}: V \to R$. For instance, in mechanics, the velocity vector space has a dual, the momentum vector space. Likewise, the tangent vector space T_pM has a dual space, namely the cotangent space T_p^*M. Thus, the tangent bundle TM has the dual, the cotangent bundle T^*M. For each p on the (base) manifold M, an element (also called a covector) in the cotangent space T_p^*M is a linear map $T_pM \to R$. If $E_i, i = 1,2,..,n$ are the coordinate bases for T_pM, the linear map induces corresponding coordinate bases $e^i, i = 1,2,..,n$ for T_p^*M such that:

$$e^j\left(E_i\right) = \delta_i^j \tag{4.16}$$

where δ_i^j is the Kronecker delta which is 1 if $i = j$ and 0 if $i \neq j$. By convention, the components of a vector in a vector space are expressed by the upper indices $v^i, i = 1,2,\ldots,n$ and the components of covectors by lower indices $w_i, i = 1,2,\ldots,n$. This is in line with the Einstein convention for summation such that $v \in T_pM$ is expressible as $v = v^iE_i$ and similarly a covector $w = w_ie^i \in T_p^*M$.

Vector fields

We are familiar with a vector field in R^n as a smooth collection of vectors (e.g. a smooth curve of vectors) assigned to a set of points in a subset R_v^n of the

n-dimensional space. That is, it is a mapping $V_f : R_v^n \to R^n$. Familiar examples are gravitational and magnetic force fields distributed in R^n. Similarly, a vector field on a manifold M is a vector-valued function X that smoothly assigns to each point $p \in M$ a tangent vector $X(p) \in T_p M$. The vector field on M may be considered as a (smooth) section of the tangent bundle TM of M. In terms of the local coordinates $\boldsymbol{x} = (x^1, x^2, .., x^n)$, we write:

$$X(p) = X^i(p) \left. \frac{\partial}{\partial x^i} \right|_p \tag{4.17}$$

where $X^i \in \mathbb{R}$. The mapping is smooth in that the components X^i may be considered as smooth functions of $\boldsymbol{x}$ in any local coordinate system induced by a chart.

Differentials, push-forwards and pull-backs

In the same way as differential forms are defined on Euclidean spaces (Edelen 1985), one can have similar definitions on manifolds (Tu 2011, Lee 2003). A differential k-form allocates to each point $p \in M$ a k-covector on the tangent space $T_p M$. For example, if a function $f \in C^\infty(M)$ which is considered a 0-form, the differential of f or the 1-form df at a point $p \in M$ is denoted by df_p such that:

$$df_p(v) = vf \,(\text{i.e. the operator } v \text{ acting on } f), \text{ for all } v \in T_p M \tag{4.18}$$

The linear transformation df_p belongs to $T_p^* M$, the dual space to $T_p M$. A covector field, a smooth section of $T^* M$, is a smooth collection (curve) of differential 1-forms and is indeed a function ω that assigns to each point $p \in M$ a covector ω_p.

Also, a smooth point map F between two manifolds induces a linear map, called its differential dF, between the tangent spaces at the corresponding points (i.e. the pre-image p and the image $F(p)$).

Thus, given two manifolds $M \subset R^n$ and $N \subset R^m$ and $F : M \to N$, one has the linear map between the tangent bundles TM and TN. Specifically, at the point p, we have a linear map called the differential $F_{*p} : T_p M \to T_{F(p)} N$. The differential F_{*p} (Figure 4.9) (may also be denoted by dF_p) is said to 'push' forward a tangent vector in $T_p M$ to a tangent vector in $T_{F(p)} N$. In terms of local coordinates, let $p \in M$ be represented by the coordinates $\boldsymbol{x} = (x^1, x^2, .., x^n)$ with $\left(\left. \frac{\partial}{\partial x^1} \right|_p \left. \frac{\partial}{\partial x^2} \right|_p \cdots \left. \frac{\partial}{\partial x^n} \right|_p \right)$ forming a basis in $T_p M$. Similarly, let $F(p) = F(x^1, x^2, .., x^n)$ be given by $\boldsymbol{y} = (y^1, y^2, .., y^m)$

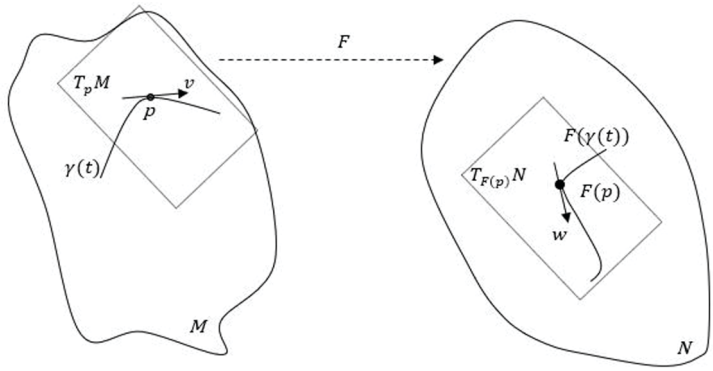


FIGURE 4.9 Differential map between two manifolds $M \subset R^n$ and $N \subset R^m; w = F_{*_p}(v)$.

with the corresponding basis $\left(\left.\frac{\partial}{\partial y^1}\right|_{F(p)} \left.\frac{\partial}{\partial y^2}\right|_{F(p)} \cdots \left.\frac{\partial}{\partial y^m}\right|_{F(p)} \right)$ in $T_{F(p)}N$. Then, the linear map F_{*_p} (push forward of the map F) is described by a matrix $[\boldsymbol{J}]_{m\times n}$ relative to these bases given by:

$$\boldsymbol{J} = \frac{\partial F^i}{\partial x^j}(p) \tag{4.19}$$

where F^i is the i^{th} component of F. Thus, $\boldsymbol{J}$ is the Jacobian matrix of the derivative of F at p and this is in line with the definition of the Jacobian matrix in the calculus of, say, any two variables. F induces a map F^* of N to M (in the opposite direction). Thus, if $g \in C^\infty(N)$ is a function (0-form) on the manifold N, h is the pull back of the function g via F^* expressible by the composition $g \circ F$ as:

$$\begin{gathered} F^*: N \to M \,|\, h(x^j) = g \circ F = g(y^i) = g(F(x^j)) \in C^\infty(M), \\ i = 1,2,..,m \text{ and } j = 1,2,\ldots,n \end{gathered} \tag{4.20}$$

Similarly, we also have the pullback of a differential 1-form at $F(p)$ in N to M. The pullback operation is denoted by $F_p^*: T_{F(p)}N \to T_pM$. Thus, if w is a 1-form on N, then F^*w is a 1-form on M, i.e.:

$$(F^*w)_p = F^*\left(w_{F(p)}\right) \tag{4.21}$$

For example, let $w = a_j(y^i)dy^j \in T_{F(p)}N, i, j = 1,2\ldots,m$ be a 1-form with a_j being the coefficients. The action of F^* on w is expressible as:

$$F^{*}w = F^{*}\left(a_j\left(\mathbf{y}\right)dy^j\right) = \left(a_j\left(F\left(\mathbf{x}\right)\right)\frac{\partial F^j\left(\mathbf{x}\right)}{\partial x^i}\right)dx^i = b_i\left(x\right)dx^i = v \in T_pM,$$

$$i = 1,2,\ldots,n \tag{4.22}$$

where $b_i\left(\mathbf{x}\right) = \left(a_j\left(F\left(\mathbf{x}\right)\right)\frac{\partial F^j\left(\mathbf{x}\right)}{\partial x^i}\right)$. We refer to Edelen (1985) for more details.

4.2.2 Riemannian Manifold and Riemannian Metric

Using the differentiable structure of a manifold M via the local Euclidean property, we have so far described concepts like tangent vectors at any generic point $p \in M$, differentials and linear maps from one manifold to another. The goal of this chapter being the establishment of a computing framework to perform optimization on manifolds, we now move on to describe a few other intrinsic properties of manifolds useful for that purpose. Towards this, one needs to know how to define on M a notion of distance and lengths of curves. Here we refer to a class of manifolds, namely Riemannian manifolds, which are endowed with a specific metric. The metric is known as the Riemannian metric denoted by g and may be considered as a function that, for each p and T_pM, assigns a smoothly varying inner product g_p. The tangent space being Euclidean and locally approximating M, the inner product $\langle -,- \rangle_p$ provides a notion of infinitesimal distance. A Riemannian manifold is often denoted by $\left(M,g\right)$.

A more intuitive understanding of the Riemannian metric is possible if we refer to the classical definition by Gauss in 1827. It is in terms of such natural invariant quantity like the length of a curve on a curved surface which is a manifold. Consider a parametrized curve $\gamma\left(t\right): R \to R^n$ lying on an m-dimensional surface ($m < n$) with $t \in \left[a,b\right]$. The surface, being embedded in an n-dimensional Euclidean space, may be represented by the Cartesian coordinates $\mathbf{x}\left(t\right) = x^1\left(t\right), x^2\left(t\right),..,x^n\left(t\right)$ as well as by the local coordinates $\mathbf{u}\left(t\right) = u^1\left(t\right), u^2\left(t\right),..,u^m\left(t\right)$ induced by a chart in the neighbourhood of a point $p = \gamma\left(t\right), a \le t \le b$ on the tangent space T_pM. At any t, the parametric surface is a vector-valued function $\mathbf{r}\left(t\right) = \left(x^1\left(\mathbf{u}\right), x^2\left(\mathbf{u}\right),\ldots,x^n\left(\mathbf{u}\right)\right)$ and the arc length of the curve over the interval $\left[a,b\right]$ is:

$$I = \int_a^b \left\| \frac{dr}{dt} \right\| dt \tag{4.23}$$

dr is an infinitesimal arc length on the curve given by:

$$dr^2 = \left(\frac{\partial r}{\partial u^k}du^k\right)^2 \tag{4.24}$$

with Einstein convention implied in the bracketed expression above. dr^2 may now be expressed as:

$$dr^2 = d\boldsymbol{u}^T \mathfrak{g}\, d\boldsymbol{u} \tag{4.25}$$

with $d\boldsymbol{u} = \left(du^1, du^2, \ldots, du^m\right)^T$. $\mathfrak{g}$ is an $m \times m$ symmetric matrix of coefficients:

$$\mathfrak{g}_{ij} = \frac{\partial \boldsymbol{r}}{\partial u^i} . \frac{\partial \boldsymbol{r}}{\partial u^j} \tag{4.26}$$

$\mathfrak{g}_{ij}$ is thus given by the dot product of the coordinate bases in the tangent space T_pM. dr^2 in Equation (4.25) is the Riemannian metric g. $\frac{\partial \boldsymbol{r}}{\partial u^i}$ is a vector and written in terms of coordinates $\boldsymbol{x}$ and $\boldsymbol{u}$, we get:

$$\frac{\partial \boldsymbol{r}}{\partial u^i} = \left(\frac{\partial x^1}{\partial u^i}, \frac{\partial x^2}{\partial u^i}, \ldots, \frac{\partial x^n}{\partial u^i} \right)^T, i = 1,2,\ldots,m \tag{4.27}$$

The Riemannian metric g is obtained as:

$$g = du^i \mathfrak{g}_{ij} du^j \tag{4.28}$$

Einstein convention is as usual implied in Equation (4.28). The metric provides the means to obtain the inner product of two vectors $\boldsymbol{v}, \boldsymbol{w} \in T_pM$ as:

$$\left\langle \boldsymbol{v}, \boldsymbol{w} \right\rangle_p = \boldsymbol{v}^T \mathfrak{g} \boldsymbol{w} = v^i \mathfrak{g}_{ij} w^j \tag{4.29}$$

Similarly, the norm of a tangent vector $\boldsymbol{v} \in T_pM$ on a Riemannian manifold is defined by $\left\| v \right\|_p = \left\langle \boldsymbol{v}, \boldsymbol{v} \right\rangle_p^{1/2} = \left(\boldsymbol{v}^T \mathfrak{g} \boldsymbol{v} \right)_p^{1/2}$. The inverse $\mathfrak{g}^{-1}$ with components denoted by $\mathfrak{g}^{ij}$ (with indices as superscripts) defines a cometric, so that the inner product of covectors $\mathcal{v}$ and $\mathcal{w}$ in the cotangent space T_p^*M is given by

$$\left\langle \mathcal{v}, \mathcal{w} \right\rangle_p = \mathcal{v}^T \mathfrak{g}^{-1} \mathcal{w} = \mathcal{v}_i \mathfrak{g}^{ij} \mathcal{w}_j \tag{4.30}$$

The inner product in Equation (4.29) is known as the first fundamental form of curved surfaces or manifolds. It is invariant under (possibly nonlinear) coordinate transformation. The importance of the first fundamental form lies in that it helps in the evaluation of the arc length of a curve on M and the angle between two parameterized

curves and areas of bounded regions without a reference to the ambient space (Do Carmo 1976).

Having defined a Riemannian metric as a scalar product on T_pM, we also note that it is a symmetric bilinear map $g_p : T_pM \times T_pM \to R$. In each chart, the metric is thus represented by a symmetric positive definite matrix $\mathfrak{g}$.

Example 4.1. The Euclidean space R^n is a trivial example of a Riemannian manifold. The tangent space T_pR^n for every p is the same R^n with $\mathfrak{g}_{ij} = \delta_{ij}$. The inner product $\langle \boldsymbol{v}, \boldsymbol{w} \rangle = v^i \delta_{ij} w^j$ is simply the dot product $\boldsymbol{v}.\boldsymbol{w}$. Thus dr^2 in Equation (4.25) is the same as the familiar metric $g = \sum_{i=1}^{n} dx_i^2$ in R^n. Note that with change of coordinates, the matrix $\mathfrak{g}_{ij}$ may be different from δ_{ij}, but the metric is unchanged. For example, in terms of polar coordinates, for $n = 2$, we have $x_1 = a\cos\theta$ and $x_2 = a\sin\theta$ with $a = \|\boldsymbol{r}\|$. Thus, $\boldsymbol{r} = \left(x_1(a,\theta), x_2(a,\theta)\right) = \left(a\cos\theta, a\sin\theta\right)$, and $\mathfrak{g}_{11} = \left(\frac{\partial r}{\partial a}.\frac{\partial r}{\partial a}\right) = 1$, $\mathfrak{g}_{12} = \left(\frac{\partial r}{\partial a}.\frac{\partial r}{\partial \theta}\right) = 0 = \mathfrak{g}_{21}$ and $\mathfrak{g}_{22} = \left(\frac{\partial r}{\partial \theta}.\frac{\partial r}{\partial \theta}\right) = a^2$. In this case, the metric $g = \begin{pmatrix} da \\ d\theta \end{pmatrix}^T \mathfrak{g} \begin{pmatrix} da \\ d\theta \end{pmatrix} = da^2 + a^2 d\theta^2$. In terms of the standard Cartesian coordinates, we have $g = dx_1^2 + dx_2^2$ which indeed gives $da^2 + a^2 d\theta^2$ when $dx_1 = \cos\theta da - a\sin\theta d\theta$ and $dx_2 = \sin\theta da + a\cos\theta\, d\theta$ are substituted.

■

Example 4.2. We consider the unit sphere $S^2 = \{x, y, z \mid x^2 + y^2 + z^2 - 1 = 0\} \subset R^3$. Let us take the chart with the local coordinates (u, v) such that $x = u$, $y = v$ and $z = +\sqrt{1 - x^2 - y^2} = +\sqrt{1 - u^2 - v^2}$ thereby projecting the upper half of the sphere on the $u - v$ plane. Here $n = 3$ and $m = 2$. With $\partial \boldsymbol{r} / \partial u = \left(\partial x / \partial u, \partial y / \partial u, \partial z / \partial u\right)^T$ and $\partial \boldsymbol{r} / \partial v = \left(\partial x / \partial v, \partial y / \partial v, \partial z / \partial v\right)^T$, the matrix elements in $\mathfrak{g}$ are given by:

$$\mathfrak{g}_{11} = \left(\frac{\partial \boldsymbol{r}}{\partial u}.\frac{\partial \boldsymbol{r}}{\partial u}\right) = \left(\begin{pmatrix} 1 \\ 0 \\ -\frac{u}{\sqrt{1-u^2-v^2}} \end{pmatrix}.\begin{pmatrix} 1 \\ 0 \\ -\frac{u}{\sqrt{1-u^2-v^2}} \end{pmatrix}\right)$$

$$= 1 + \frac{u^2}{1-u^2-v^2}$$

$$\mathfrak{g}_{12} = \left(\frac{\partial \boldsymbol{r}}{\partial u}.\frac{\partial \boldsymbol{r}}{\partial v}\right) = \left(\begin{pmatrix} 1 \\ 0 \\ -\dfrac{u}{\sqrt{1-u^2-v^2}} \end{pmatrix}.\begin{pmatrix} 0 \\ 1 \\ -\dfrac{v}{\sqrt{1-u^2-v^2}} \end{pmatrix}\right) = \frac{uv}{1-u^2-v^2} = \mathfrak{g}_{21} \text{ and}$$

$$\mathfrak{g}_{22} = \left(\frac{\partial \boldsymbol{r}}{\partial v}.\frac{\partial \boldsymbol{r}}{\partial v}\right) = \left(\begin{pmatrix} 1 \\ 0 \\ -\dfrac{v}{\sqrt{1-u^2-v^2}} \end{pmatrix}.\begin{pmatrix} 1 \\ 0 \\ -\dfrac{v}{\sqrt{1-u^2-v^2}} \end{pmatrix}\right) = 1 + \frac{v^2}{1-u^2-v^2}$$

(4.31a,b,c)

We get the matrix $\mathfrak{g}$ as:

$$\mathfrak{g} = \begin{bmatrix} \dfrac{1-v^2}{1-u^2-v^2} & \dfrac{uv}{1-u^2-v^2} \\ \dfrac{uv}{1-u^2-v^2} & \dfrac{1-u^2}{1-u^2-v^2} \end{bmatrix} \tag{4.31d}$$

Equation (4.28) gives the metric g as $\begin{pmatrix} du & dv \end{pmatrix}\ \mathfrak{g}\ \begin{pmatrix} du & dv \end{pmatrix}^T$, i.e.:

$$\begin{aligned} g &= \mathfrak{g}_{11}du^2 + \mathfrak{g}_{12}dudv + \mathfrak{g}_{21}dvdu + \mathfrak{g}_{22}dv^2 \\ &= \left(1 + \frac{u^2}{1-u^2-v^2}\right)du^2 + \frac{2uv}{1-u^2-v^2}dudv + \left(1 + \frac{v^2}{1-u^2-v^2}\right)dv^2 \end{aligned} \tag{4.32}$$

■

4.2.3 Geodesic on a Manifold

As mentioned in the introduction to this chapter, a geodesic $\gamma(t)$, say with $t \in [a,b]$, obtains the shortest path between two points $\gamma(a)$ and $\gamma(b)$ on a manifold. In the Euclidean space, it is simply the length of the straight line segment between the points. If a straight line is represented in R^n by the curve $\gamma(t) = \alpha + \beta t$ with $\alpha, \beta \in R$, we find that a straight line is a curve with $\ddot{\gamma} = 0$. If one wishes to adopt a similar definition for a geodesic on a manifold, we face a difficulty. With the curve $\gamma(t)$ lying on a manifold M, the tangent vectors $\dot{\gamma}(t_1)$ and $\dot{\gamma}(t_1 + \Delta t)$ lie in different tangent spaces $T_{\gamma(t_1)}M$ and $T_{\gamma(t_1+\Delta t)}M$. Therefore, it is inappropriate to define a difference ratio $\dfrac{\left(\dot{\gamma}(t_1 + \Delta t) - \dot{\gamma}(t_1)\right)}{\Delta t}$ before taking the limit as $\Delta t \to 0$ and make an attempt to

build the acceleration vector $\ddot{\gamma}(t_1)$. Yet it is possible to connect adjacent tangent spaces in a manifold and define an acceleration operation via the important concept of 'connection' in the theory of manifolds. Before we dwell upon this concept and elaborate it in the next section, let us derive the geodesic equation for manifolds by the very definition of a geodesic being the shortest path between two points.

We start with Equation (4.23) wherein the integral gives the arc length of the curve $\gamma(t)$ on the manifold as t varies over the interval $[a,b]$. The task being to find such a curve that minimizes the arc length among many other possible curves joining the two points, it is indeed a variational problem (see Section 1.5.1, Chapter 1) leading to the minimization of a functional. Following the procedure described therein, we have here the integral I in Equation (4.23) as the functional and the integrand, denoted by L, may be taken without a loss of generality as $\left(\frac{d\boldsymbol{r}}{dt}\right)^2$ instead of the norm $\left\|\frac{d\boldsymbol{r}}{dt}\right\|$. Thus:

$$L = \mathfrak{g}_{ij}\frac{du^i}{dt}\frac{du^j}{dt} = \mathfrak{g}_{ij}\dot{u}^i\dot{u}^j, i, j = 1,2,\ldots,m \tag{4.33}$$

Keeping in mind the Einstein convention in the expression on the RHS of the equation above, we obtain the Euler-Lagrange equations in the form:

$$\frac{\partial L}{\partial u^k} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{u}^k}\right) = 0, k = 1,2,\ldots,m \tag{4.34}$$

Noting that $\mathfrak{g}_{ij}$ is a function of u^i and u^j, we have:

$$\frac{\partial L}{\partial u^k} = \frac{\partial \mathfrak{g}_{ij}}{\partial u^k}\dot{u}^i\dot{u}^j \tag{4.35}$$

$$\begin{aligned}\frac{\partial L}{\partial \dot{u}^k} &= \mathfrak{g}_{ij}\frac{\partial \dot{u}^i}{\partial \dot{u}^k}\dot{u}^j + \mathfrak{g}_{ij}\dot{u}^i\frac{\partial \dot{u}^j}{\partial \dot{u}^k} \\ &= \mathfrak{g}_{ij}\delta^i_k\dot{u}^j + \mathfrak{g}_{ij}\dot{u}^i\delta^j_k \\ &= \mathfrak{g}_{kj}\dot{u}^j + \mathfrak{g}_{ik}\dot{u}^i\end{aligned} \tag{4.36}$$

and

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{u}^k}\right) = \frac{d}{dt}\left(\mathfrak{g}_{kj}\dot{u}^j + \mathfrak{g}_{ik}\dot{u}^i\right)$$

$$= \frac{d\mathfrak{g}_{kj}}{dt}\dot{u}^j + \mathfrak{g}_{kj}\ddot{u}^j + \frac{d\mathfrak{g}_{ik}}{dt}\dot{u}^i + \mathfrak{g}_{ik}\ddot{u}^i$$

$$= \frac{\partial \mathfrak{g}_{kj}}{\partial u^i}\dot{u}^i\dot{u}^j + \mathfrak{g}_{kj}\ddot{u}^j + \frac{\partial \mathfrak{g}_{ik}}{\partial u^j}\dot{u}^j\dot{u}^i + \mathfrak{g}_{ik}\ddot{u}^i \tag{4.37a}$$

Renaming the dummy variables and using the symmetry of the matrix $\mathfrak{g}$, we simplify Equation (4.37a) as:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{u}^k}\right) = \left(\frac{\partial \mathfrak{g}_{kj}}{\partial u^i} + \frac{\partial \mathfrak{g}_{ik}}{\partial u^j}\right)\dot{u}^j\dot{u}^i + 2\mathfrak{g}_{ik}\ddot{u}^i \tag{4.37b}$$

Substituting Equations (4.35) and (4.37b) in Equation (4.34) gives:

$$\frac{\partial \mathfrak{g}_{ij}}{\partial u^k}\dot{u}^i\dot{u}^j - \left(\left(\frac{\partial \mathfrak{g}_{kj}}{\partial u^i} + \frac{\partial \mathfrak{g}_{ik}}{\partial u^j}\right)\dot{u}^j\dot{u}^i + 2\mathfrak{g}_{ik}\ddot{u}^i\right) = 0$$

$$\Rightarrow \ddot{u}^k = -\Gamma^k_{ij}\dot{u}^i\dot{u}^j, k = 1,2,\ldots,m \tag{4.38}$$

where Γ are known as the Christoffel symbols (Christoffel 1869, Lee 1997) given by:

$$\Gamma^k_{ij} = \frac{1}{2}\mathfrak{g}^{kl}\left(\frac{\partial \mathfrak{g}_{jl}}{\partial u^i} + \frac{\partial \mathfrak{g}_{il}}{\partial u^j} - \frac{\partial \mathfrak{g}_{ij}}{\partial u^l}\right) \tag{4.39}$$

Equation (4.38) is the geodesic equation.

In the Euclidean case, $\mathfrak{g}_{ij} = \delta_{ij}$ and therefore the Christoffel symbols are identically zero. Hence, we have $\ddot{u}^k = 0$ with the only solution being a straight line as expected. A few examples highlighting the geodesic curves on manifolds may be apt at this stage. This, in general, requires the integration of Equation (4.38).

Example 4.3. For the sphere $S^2 \subset R^3$, the great circles are the geodesics.

Solution. To prove this statement, let us consider a sphere with radius a and form the $\mathfrak{g}$ matrix using spherical coordinates ϕ and θ. Any point on the sphere is represented by $x = a\sin\phi\cos\theta$, $y = a\sin\phi\sin\theta$ and $z = a\cos\phi$ with $\phi \in [0,\pi]$ being the polar angle and $\theta \in [0,2\pi)$ the azimuth angle (Figure 4.10). The coefficients of $\mathfrak{g}$ are:

$$\mathfrak{g}_{11} = \left(\frac{\partial \boldsymbol{r}}{\partial \phi}\cdot\frac{\partial \boldsymbol{r}}{\partial \phi}\right) = a^2, \mathfrak{g}_{12} = \left(\frac{\partial \boldsymbol{r}}{\partial \phi}\cdot\frac{\partial \boldsymbol{r}}{\partial \theta}\right) = 0 = \mathfrak{g}_{21}, \mathfrak{g}_{22} = \left(\frac{\partial \boldsymbol{r}}{\partial \theta}\cdot\frac{\partial \boldsymbol{r}}{\partial \theta}\right) = a^2\sin^2\phi$$

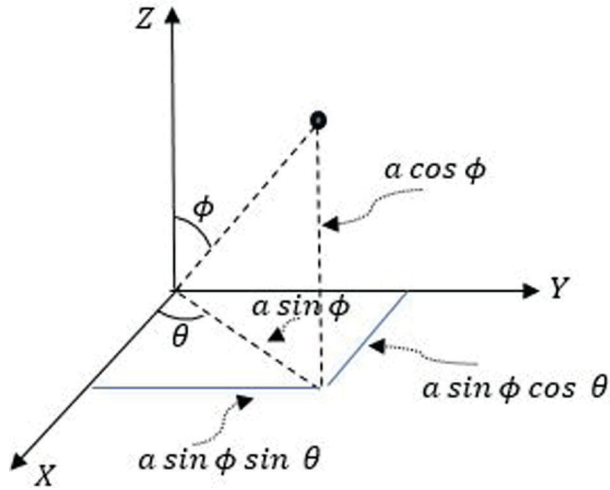


FIGURE 4.10 A sphere $S^2 \subset R^3$; spherical coordinate system.

$$\mathfrak{g} = \begin{bmatrix} a^2 & 0 \\ 0 & a^2 \sin^2 \phi \end{bmatrix} \tag{4.40a}$$

Its inverse is:

$$\mathfrak{g}^{-1} = \begin{bmatrix} \dfrac{1}{a^2} & 0 \\ 0 & \dfrac{1}{a^2}\operatorname{cosec}^2 \phi \end{bmatrix} \tag{4.40b}$$

The Christoffel symbols are $\Gamma_{ij}^k = \frac{1}{2}\mathfrak{g}^{kl}\left(\frac{\partial \mathfrak{g}_{jl}}{\partial u^i} + \frac{\partial \mathfrak{g}_{il}}{\partial u^j} - \frac{\partial \mathfrak{g}_{ij}}{\partial u^l}\right),\ i,j,k,l = 1,2.$ These are evaluated in Appendix 4 (item A4.1) in terms of the spherical coordinates for a sphere with unit radius ($a = 1$). From Equation (4.38), the geodesic equations for ϕ and θ are given by:

$$\begin{aligned} \ddot{\phi} &= -\Gamma_{11}^1 \dot{\phi}^2 - \left(\Gamma_{12}^1 + \Gamma_{21}^1\right)\dot{\phi}\dot{\theta} - \Gamma_{22}^1 \dot{\theta}^2 \\ \ddot{\theta} &= -\Gamma_{11}^2 \dot{\phi}^2 - \left(\Gamma_{12}^2 + \Gamma_{21}^2\right)\dot{\phi}\dot{\theta} - \Gamma_{22}^2 \dot{\theta}^2 \end{aligned} \tag{4.41}$$

Inserting the expressions for the Christoffel symbols, Equation (4.41) takes the form:

$$\ddot{\phi} = \sin\phi\cos\phi\dot{\theta}^2$$

$$\ddot{\theta} = -2\operatorname{cosec}\phi\cos\phi\dot{\phi}\dot{\theta} \tag{4.42a,b}$$

From the two ODEs above, we readily infer the following. For a constant θ, Equation (4.42b) is trivially satisfied and Equation (4.42a) gives $\ddot{\phi} = 0$ yielding the result $\phi(t) = at + b$ with $a, b \in R$. This shows that the geodesic lies on all great circles (passing through the two poles) for any specific azimuth angle $\theta \in [0, 2\pi)$.

On the other hand, if ϕ is constant, we get from Equations (4.42) the following ODEs:

$$\sin\phi\cos\phi\dot{\theta}^2 = 0$$

$$\ddot{\theta} = 0 \tag{4.43a,b}$$

While θ may vary linearly with the parameter t, the constant values that ϕ may assume are given by:

$$\sin\phi_c\cos\phi_c\dot{\theta}^2 = 0 \tag{4.44}$$

With θ varying with t and $\dot{\theta}$ being constant, ϕ_c may assume the values $0, \frac{\pi}{2}, \pi$.

However, the only plausible value for ϕ_c is $\frac{\pi}{2}$ corresponding to the equator which is also a great circle.

■

Example 4.4. For the same unit sphere $S^2 \subset R^3$, with the metric g given by Equation (4.32) in terms of Cartesian coordinates u and v and the coefficient matrix $\mathfrak{G}$ as derived in Equation (4.31), we numerically solve the geodesic equation (4.38).

Solution. The solution to the geodesic Equation (4.38) is obtained by numerical integration. It involves solving two coupled second-order nonlinear ODEs in terms of u and v. Here the equations are solved as an initial value problem with specified initial conditions (ICs) $u_0, \dot{u}_0, v_0$ and $\dot{v}_0$. Figure 4.11 shows the geodesics obtained with two different sets of initial conditions. The first set corresponds to the north pole $(u_0 = 0,\ v_0 = 0)$ with assumed velocities $\dot{u}_0 = 0.1$ and $\dot{v}_0 = 0$. The second set are the initial conditions $u_0 = 0.3, \dot{u}_0 = 0.4, v_0 = 0.4$ and $\dot{v}_0 = -0.3$. The end point in both the cases is taken as that obtained after $t = 5$ s. Note that with any choice of the initial point, the integration needs to be performed such that the final point lies within the one half (hemisphere) of S^2. This is in view of the fact that S^2 is fully represented by two coordinate charts, one covering the upper half with the initial point (u_0, v_0) considered as a pole and the other the lower half. In any case, solutions for the geodesic always lie on great circles as also evident from the earlier example.

■

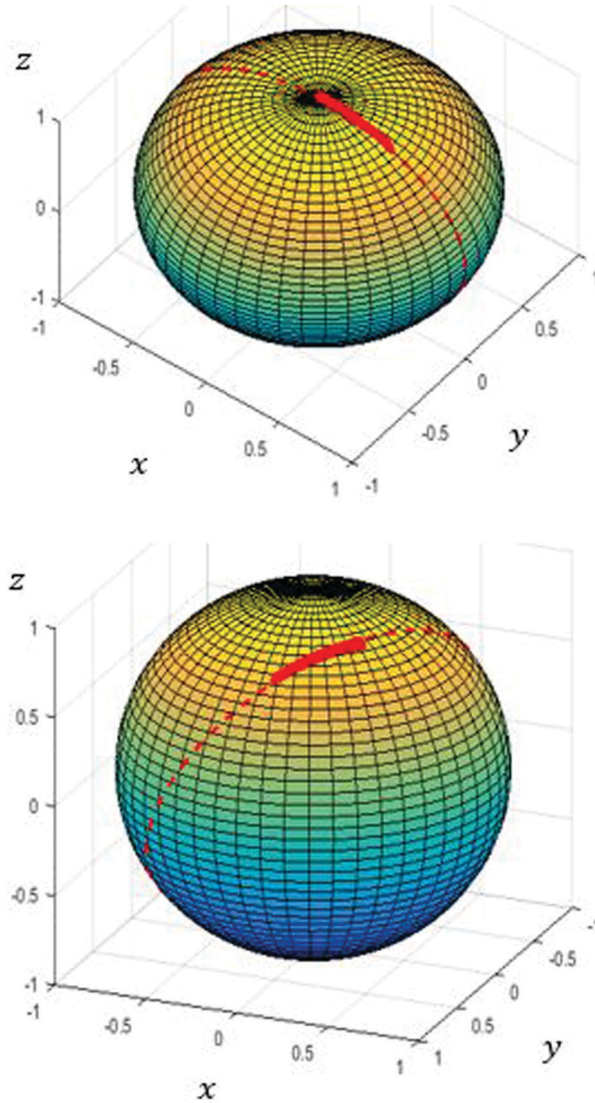


FIGURE 4.11 Geodesics (in solid line) for unit sphere $S^2 \subset R^3$: (a) ICs $u_0 = 0, \dot{u}_0 = 0.1, v_0 = 0$ and $\dot{v}_0 = 0$, (b) ICs $u_0 = 0.3, \dot{u}_0 = 0.4, v_0 = 0.4$ and $\dot{v}_0 = -0.3$, dashed circle in line corresponds to a great circle of which the geodesic forms a segment of minimum distance between its end points.

4.2.4 Connection on a Manifold and Covariant Derivative

As noted in the previous section, geodesics are the Riemannian generalization of straight lines of Euclidean geometry. This definition implies that geodesics are indeed curves of zero acceleration, i.e., the unit tangent vector is constant as we move along

a geodesic curve. In order to adopt this property of a geodesic on a manifold M, we need to measure the variation in the tangent vectors between two adjacent points on M. The notion of connection which we describe here connects the tangent spaces of manifolds and helps us to define directional derivatives of vector fields (called the covariant derivatives) along curves. This property we utilize in establishing the fact that the geodesic is a curve on M with zero tangential acceleration.

With vector fields $X, Y \in TM$, a connection on a (Riemannian) manifold M is defined by a map:

$$\nabla : TM \times TM \to TM \tag{4.45}$$

written as $(X, Y) \mapsto \nabla_X Y$. It satisfies the properties:

(i) $\nabla_X Y$ is linear over $C^\infty(M)$ in X, i.e.:

$$\nabla_{fX_1 + gX_2} Y = f\nabla_{X_1} Y + g\nabla_{X_2} Y \text{ with } f, g \in C^\infty(M) \tag{4.46a}$$

(ii) $\nabla_X Y$ is linear over R in Y, i.e:

$$\nabla_X (aY_1 + bY_2) = a\nabla_X Y_1 + b\nabla_X Y_2 \text{ with } a, b \in R \tag{4.46b}$$

(iii) $\nabla_X (fY) = f\nabla_X Y + X(f)Y,\ f \in C^\infty(M)$ (4.46c)

The connection is equivalently specified by a covariant derivative, an operator that differentiates sections of a tangent bundle along any tangent direction in the manifold M. For a better understanding of the connection operator ∇, let us express $\nabla_X Y$ in terms of local coordinates $\boldsymbol{u}(t) = u^1(t), u^2(t), .., u^m(t)$ induced by a chart (U, φ). The basis vectors $E_j = \dfrac{\partial}{\partial u^j}, j = 1, 2, \ldots, m$ span the tangent space $T_p M$ at each $p \in U$ and may be treated as vector fields. In this context, one fundamental result is that the covariant derivative $\nabla_{E_i} E_j$ is expressible in terms of the Christoffel symbols (Lee 1997) as:

$$\nabla_{E_i} E_j = \Gamma_{ij}^k E_k \tag{4.47}$$

When vector fields X and Y are expressed in terms of the local basis vectors as $X = X^i E_i$ and $Y = Y^j E_j$, then:

$$\begin{aligned}
\nabla_X Y &= \nabla_X Y^j E_j \\
&= \left(X^i E_i Y^j\right) E_j + Y^j \nabla_{X^i E_i} E_j \left(\text{from Equation 4.46c}\right) \\
&= \left(X^i E_i Y^j\right) E_j + X^i Y^j \nabla_{E_i} E_j \left(\text{from Equation 4.46a}\right)
\end{aligned}$$

$$= \left(X^i E_i Y^j \right) E_j + X^i Y^j \Gamma_{ij}^k E_k \left(\text{from Equation 4.47} \right)$$

$$= \left(X^i E_i Y^k + X^i Y^j \Gamma_{ij}^k \right) E_k \left(\text{since '} j \text{ is a dummy variable} \right) \tag{4.48}$$

That Christoffel symbols Γ_{ij}^k completely define the action of connection on manifolds is evident from the above equation.

Covariant derivatives along curves

We consider the parameterized curve $\gamma(t) : \mathbb{R} \to M$ on the manifold M with t varying over the interval $I = [a, b] \subset \mathbb{R}$. Let the local coordinates be $\boldsymbol{u}(t) = u^1(t), u^2(t), .., u^m(t)$ induced by a chart (U, φ) in the neighbourhood U of a point $p = \gamma(t)$ on the tangent space $T_p M$. The tangent vector $\dot{\gamma}(t) = \left(\dot{u}^1(t), \dot{u}^2(t), .., \dot{u}^m(t) \right) \in T_p M$ is coordinate chart invariant as discussed in Section 4.2.2. It may not however be so for the acceleration vector $\ddot{\gamma}(t)$. Using the fact that the tangent vector $\dot{\gamma}(t) = \dot{u}^i(t) E_i$ is indeed a vector field along the curve $\gamma(t)$, we proceed to know via the covariant derivative how the tangent space $T_p M$ changes along $\gamma(t)$. Consider a vector field V along the curve $\gamma(t)$ such that $V(t) \in T_{\gamma(t)} M \; \forall t \in \mathbb{R}$. Suppose that $\mathbb{T}(\gamma)$ is the space of vector fields along the curve $\gamma(t)$; we define the linear connection ∇ on M for each curve $\gamma(t)$ as a unique differential operator $D_t : \mathbb{T}(\gamma) \to \mathbb{T}(\gamma)$ satisfying the properties:

(i) $D_t(aV + bW) = aD_t V + bD_t W$, for $a, b \in \mathbb{R}$– linearity over $\mathbb{R}$ with V and W being vector fields

(ii) $D_t(fV) = \dot{f} V + f D_t V$, for $f \in C^\infty(I)$ – product rule (4.49)

$D_t V$ may be called the covariant derivative of V along the curve $\gamma(t)$. With the acceleration of the curve $\gamma(t)$ given by $D_t \dot{\gamma}(t)$, we can say that $\gamma(t)$ is a geodesic if $D_t \dot{\gamma} = 0$. This condition is given by:

$$D_t \dot{\gamma} = \nabla_{\dot{\gamma}(t)} \dot{\gamma} = \nabla_{\dot{u}^i(t) E_i} \left(\dot{u}^j(t) E_j \right) = 0$$

$$\Rightarrow \left(\dot{u}^i E_i \dot{u}^k + \dot{u}^i \dot{u}^j \Gamma_{ij}^k \right) E_k = 0 \left(\text{from Equation 4.48} \right)$$

$$\Rightarrow \left(\ddot{u}^k + \dot{u}^i \dot{u}^j \Gamma_{ij}^k \right) E_k = 0 \tag{4.50}$$

The last equation indicates that the curve $\gamma(t)$ is a geodesic if the component functions of $\gamma(t) = \left(u^1(t), u(t), .., u^m(t) \right)$ satisfy the ODEs:

$$\ddot{u}^k + \dot{u}^i \dot{u}^j \Gamma_{ij}^k = 0, \; k = 1, 2, \ldots, m \tag{4.51a}$$

These ODEs are the same as equations (4.38) earlier derived in Section 4.2.3 by the variational approach. One may also rewrite these as $2m$ first-order ODEs:

$$\dot{u}^k = w^k$$

$$\dot{w}^k = -w^i w^j \Gamma^k_{ij} \tag{4.51b,c}$$

The terms on the RHS being C° functions of $(u,w) = (u^1,u^2,\ldots,u^m,w^1,w^2,\ldots,w^m)$, the existence theorem (Boothby 1975, Rudin 1976) for ODEs ensures that there are $2m$ unique functions (solutions) for u^k and w^k satisfying the system of Equations (4.51b,c) and the specified initial conditions.

4.2.5 Parallel Transport of a Vector Field along a Curve $\gamma(t)$

Taking cue from the understanding of a covariant derivative along a curve $\gamma(t)$, we may as well deduce the condition for the parallel transport of a vector field along the curve on a manifold. A vector field V along a curve $\gamma(t)$ is said to be parallel (Figure 4.12) if the directional derivative $\nabla_{\dot{\gamma}(t)} V$ of V along the tangential direction of $\gamma(t)$ is zero at all points on the curve. With $V = v^k(t) E_k$, it readily follows that V is parallel along the curve if and only if its components $v^k(t)$ satisfy the following linear ODEs (similar to Equation 4.51c):

$$\dot{v}^k + v^i \dot{\gamma}(t)^j \Gamma^k_{ij} = 0, k = 1,2,\ldots,m \tag{4.52}$$

The Christoffel symbols Γ^k_{ij} in the above equation correspond to the point $\gamma(t)$. Thus, a parallel vector field is uniquely defined by its initial position V_{t_0}, i.e. by $v^i(t_0), i = 1,2,\ldots,m$. In Euclidean space with canonical flat connection, i.e., with vanishing Christoffel symbols, parallel transport just means that

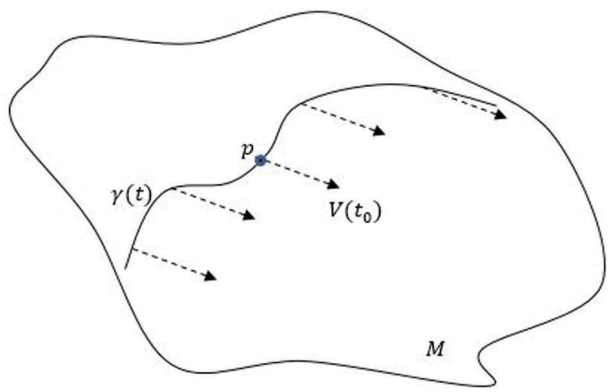


FIGURE 4.12 Parallel transport of a vector field along a curve $\gamma(t)$ on a manifold M.

$\frac{dV}{dt} = \nabla_{\dot{\gamma}(t)} V = 0 \Rightarrow \dot{v}^k(t) = 0, k = 1,2,\ldots,m,$ and so V is a constant vector. Also note here that a curve $\gamma(t)$ on a manifold is called a geodesic if the parallel transport of the tangent vector along the curve preserves the vector, i.e., a tangent vector when parallelly transported remains a tangent vector at all points on the curve.

4.2.6 Levi-Civita Connection

The Levi-Civita connection is a unique connection defined on a Riemannian manifold (M, g) having the following properties in addition to those listed in Equation (4.46):

(i) it is compatible with the Riemannian metric g and
(ii) it is a torsion-free connection

Compatibility property of Levi-Civita connection
Property (i) is expressed as:

$$\nabla_X \langle Y, Z \rangle = \langle \nabla_X Y, Z \rangle + \langle Y, \nabla_X Z \rangle \tag{4.53a}$$

X, Y, Z are vector fields on M. The property is just like a Euclidean connection (denoted by, say, ∇^E on $\mathbb{R}^n$) satisfying $\nabla^E \langle X, Y \rangle = \langle \nabla^E X, Y \rangle + \langle X, \nabla^E Y \rangle$ with respect to the Euclidean metric. The compatibility property is equivalent to saying that if V and W are vector fields along any curve $\gamma(t)$, then:

$$\frac{d}{dt}\langle V, W \rangle = \langle D_t V, W \rangle + \langle V, D_t W \rangle \tag{4.53b}$$

Proof:
With $V = v^i E_i$ and $W = w^i E_i$, we have:

$$\langle V, W \rangle = g(V, W) = v^i w^j g(E_i, E_j) = v^i w^j g_{ij} \tag{4.54a}$$

and:

$$\frac{d}{dt}\langle V, W \rangle = \frac{d}{dt} g(V, W) = \left(v^i \frac{dw^j}{dt} + w^j \frac{dv^i}{dt} \right) g_{ij}$$

$$= g\left(v^i E_i, \frac{d\left(w^j E_j\right)}{dt} \right) + g\left(\frac{d\left(v^i E_i\right)}{dt}, w^j E_j \right)$$

$$= g(V, D_t W) + g(D_t V, W) = \langle V, D_t W \rangle + \langle D_t V, W \rangle \tag{4.54b}$$

which is the same as Equation (4.53b).

♦

Now, suppose that V and W are parallel vector fields along the curve. Then $D_t V = 0 = D_t W$. We get from Equation (4.54b):

$$\frac{d}{dt}\langle V,W\rangle = 0 \Rightarrow \frac{d}{dt} g(V,W) = \nabla g = 0$$

$$\Rightarrow g\left(V_{t_0}, W_{t_0}\right) = g\left(V_{t_1}, W_{t_1}\right) \text{where } t_0, t_1 \in [a,b]. \tag{4.55}$$

Thus, the compatibility condition implies that $\langle V,W\rangle$ is constant if V and W are parallel vector fields. That is, the parallel transport preserves the metric.

Torsion-free[‡] *property (ii) of Levi-Civita connection*
Levi-Civita connection is characterized by the vanishing of the torsion tensor τ, which may be expressed in terms of the vector fields X and Y as:

$$\tau(X,Y) = \nabla_X Y - \nabla_Y X - [X,Y] = 0 \tag{4.56}$$

$[X,Y]$ on the RHS of Equation (4.56) is a bracket operation on vector fields and is known as the Lie bracket.[§] The Lie bracket of two vectors measures the failure to close the flow lines of these vectors (see the comments below).

[‡] Torsion
Torsion describes the twisting of a vector field when it is parallelly transported along a geodesic.

[§] Lie bracket
Given vector fields X and Y on the manifold M, the Lie bracket $[X,Y]$ is a commutator:

$$[X,Y] := X \circ Y - Y \circ X : C^\infty(M) \to C^\infty(M) \tag{i}$$

Even though the composition $X \circ Y$ is not a vector field (because it involves second-order derivatives), the commutator is a vector field. It is a derivative operation and finds the change of the vector field X along the vector field Y. If $f \in C^\infty(M)$ is regarded as a function in the neighbourhood of $p \in M$, then:

$$X \circ Y(f) = x^j \frac{\partial}{\partial x_j}\left(y^i \frac{\partial f}{\partial x_i}\right) = x^j \frac{\partial y^i}{\partial x_j}\frac{\partial f}{\partial x_i} + x^j y^i \frac{\partial^2 f}{\partial x_j \partial x_i} \tag{ii}$$

With a similar expression for $Y \circ X(f)$. we get:

$$[X,Y] = \left(x^j \frac{\partial y^i}{\partial x_j} - y^j \frac{\partial x^i}{\partial x_j}\right)\frac{\partial}{\partial x_i} \tag{iii}$$

which is indeed a vector field on M. One may observe that $[X,X] = 0$ and $[Y,X] = -[X,Y]$. Also:

$$[[X,Y],Z] + [[Y,Z],X] + [[Z,X],Y] = 0 \,(\text{Jacobi identity}) \tag{iv}$$

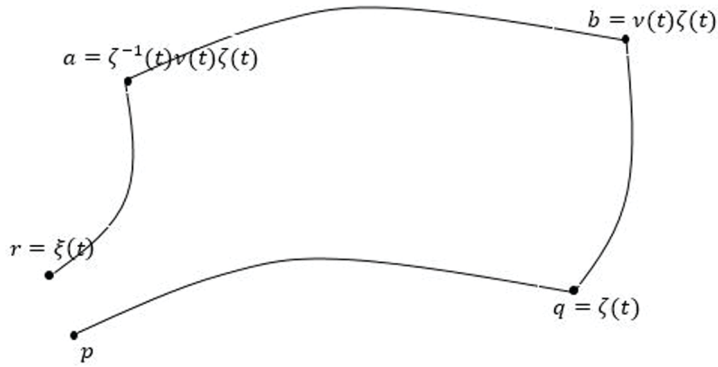


FIGURE 4.13 Geometric interpretation of Lie bracket.

Figure 4.13 illustrates the geometric interpretation of a Lie bracket. Consider two vector fields X and Y. Let ζ denote the flow of X and v the flow of Y. Starting from a point $p \in M$, we travel first along ζ, the flow field of X for a time t to reach the point q. The point b is subsequently reached by following v, the flow field of Y. Now, a movement along $\zeta^{-1}(t)$, i.e., the reverse direction of the flow field of X reaches the point a and via a similar movement along $v^{-1}(t)$, the point $r = \xi(t)$ is finally reached. Thus, one may define $\xi(t)$ (shown in the figure) as: $\xi(t) = v^{-1}(t) \circ \zeta^{-1}(t) \circ v(t) \circ \zeta(t)$.

Since ζ and v are smooth, ξ is smooth too. As $t \to 0$, the flow line joining p and r gives the direction of the vector $[X,Y]$. $\xi(t)$ as $t \to 0$ may be considered as a measure of non-commutativity of the flows ζ and v, represented by the Lie bracket $[X,Y]$. With $X = x^k \dfrac{\partial}{\partial x^k} = x^k E_k$ and $Y = y^k \dfrac{\partial}{\partial x^k} = y^k E_k$ written in terms of local coordinates, the vector field $[X,Y]$ is:

$$[X,Y] = \left(x^i E_i y^k - y^i E_i x^k\right) \frac{\partial}{\partial x^k} = \left(x^i E_i y^k - y^i E_i x^k\right) E_k \tag{4.57}$$

On the other hand, the expression $\nabla_X Y - \nabla_Y X$ in Equation (4.56) is:

$$\nabla_X Y - \nabla_Y X = \left(x^i E_i y^k + x^i y^j \Gamma_{ij}^k\right) E_k - \left(y^i E_i x^k + y^i x^j \Gamma_{ij}^k\right) E_k$$

$$= \left(x^i E_i y^k - y^i E_i x^k\right) E_k + \left(x^i y^j \Gamma_{ij}^k - y^j x^i \Gamma_{ji}^k\right) E_k \tag{4.58}$$

Therefore, from Equations (4.57) and (4.58):

$$\tau(X,Y) = \nabla_X Y - \nabla_Y X - [X,Y] = \left(x^i y^j \Gamma_{ij}^k - y^j x^i \Gamma_{ji}^k\right) E_k \tag{4.59}$$

It shows that the connection becomes torsion-free, i.e., $\tau(X,Y)=0$ if:

$$\Gamma_{ij}^{k}=\Gamma_{ji}^{k} \tag{4.60}$$

So, the Christoffel symbols are symmetric (in the two lower indices) rendering the Levi-Civita connection symmetric. Thus, on a Riemannian manifold (M,g), the Levi-Civita connection is the unique torsion-free (symmetric) connection which is compatible with the Riemannian metric. It is named after Tullio Levi-Civita who introduced the concept of parallelism on a Riemannian manifold in his famous paper published in 1917. The paper is well-known for the geometric interpretation of the Riemannian curvature and the parallel transport of vector fields on a Riemannian manifold.

4.2.7 Exponential and Logarithmic Maps

A geodesic, being the shortest path (smooth curve) between two points p, $q\in M$ and uniquely determined by a tangent vector $v\in T_pM$, is useful for movement on M. In this context, an exponential map denoted by $Exp_p(v)$ is defined as a map $Exp_p(v):T_pM\to M$ such that there is a geodesic $\gamma(t)$ with $\gamma(0)=p\in M,\ \dot{\gamma}(0)=v$ and $\gamma(t)=q=Exp_p(vt)\in M$. Specifically, for $t\in[0,1]$, $\gamma(t)$ is a geodesic. The exponential map has an inverse $Exp_p^{-1}(q):M\to T_pM$, i.e., $v=Exp_p^{-1}(q)$. The geodesic distance between p and q is $\left\|Exp_p^{-1}(q)\right\|=\left\|Exp_q^{-1}(p)\right\|=\|v\|$. Thus, $Exp_p(v)$ is the point on the geodesic whose distance from p along the geodesic is the length of the vector v. The inverse exponential map is known as the logarithmic map. See Figure 4.14 for an illustration of exponential and logarithmic maps on M.

4.2.8 Normal Coordinates

By the definition of an exponential map, we may have a system of natural local coordinates known as normal coordinates in the neighbourhood U of $p\in M$. The coordinate system has a special significance (Hsu 2002) when one deals with the topic of stochastic development on manifolds. Under exponential mapping, every point p of M has a neighbourhood U which is the diffeomorphic image of a star-shaped neighbourhood $U'\subset T_p(M)$ (Figure 4.15). The coordinate neighbourhood $U\subset M$ defined in this way is known as the normal neighbourhood. If $E_i=\dfrac{\partial}{\partial x^i}$ is an orthonormal basis in $T_p(M)$ with an isomorphism** $G:\mathbb{R}^m\to T_p(M)$, then we have a coordinate chart:

** *Isomorphism*
Isomorphism is a bijective morphism. The word *iso* derived from Greek means 'equal'. The word *morphosis* means 'to form' or 'to shape'. Isomorphism is a one-to-one mapping that preserves some structural aspects of the two mathematical objects that are mapped. For instance, if G and H are two graphs, G is isomorphic to H, denoted by $G\cong H$ if (i) their number of components (vertices and edges) are the same and (ii) their edge connectivity is retained.

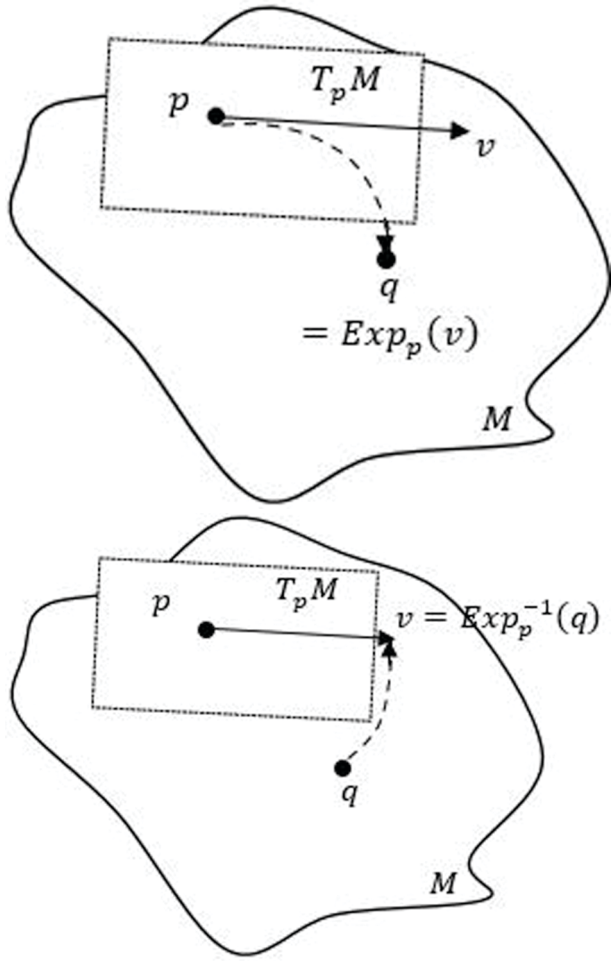


FIGURE 4.14 Manifold M: (a) exponential map $q = Exp_p(v)$; (b) logarithmic map $Exp_p^{-1}(q) = v$.

$$\varphi := G^{-1} \circ Exp_p^{-1} : U \to \mathbb{R}^m \tag{4.61}$$

The coordinates corresponding to the above map are called the normal coordinates centred at p and there is a one-to-one correspondence between the orthonormal bases at p and the normal coordinate chart φ.

Some of the properties of the normal coordinate chart are the following.

(i) For any $\boldsymbol{v} \in T_p(M)$, the geodesic $\gamma(t)$ starting at $p \in M$ with initial vector $\boldsymbol{v}$ is represented in the normal coordinate chart by the radial line segment (see Figure 4.15):

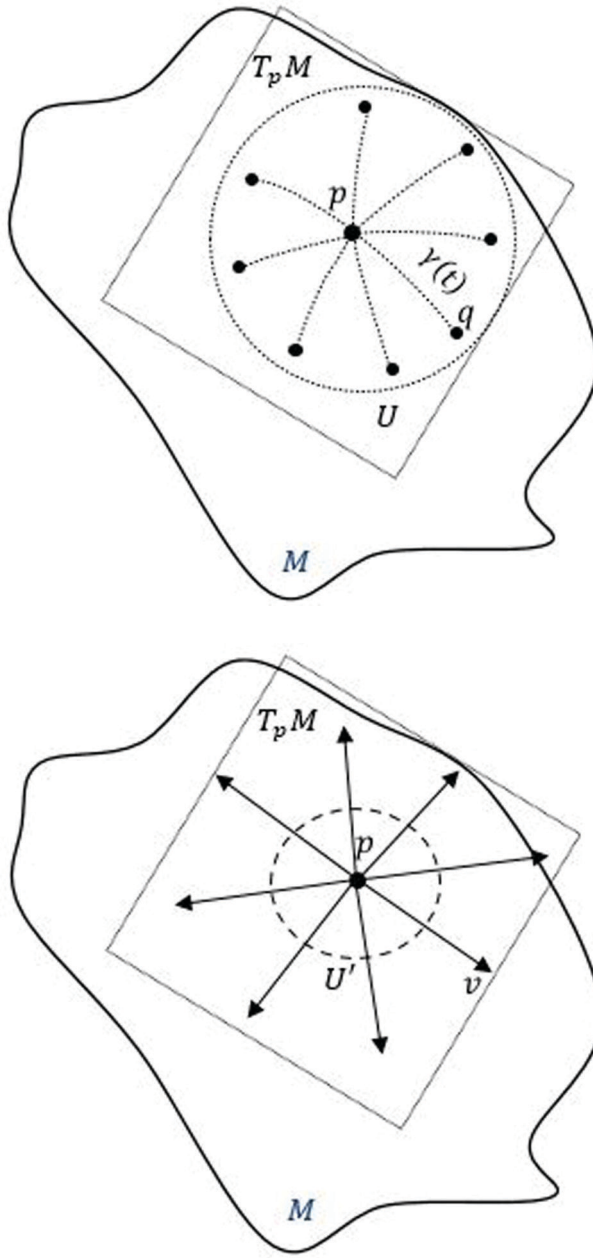


FIGURE 4.15 Normal coordinates using exponential map on a Riemannian manifold (M,g) with $v \in T_p(M)$: (a) $U \subset M$ – the diffeomorphic image of (b) a star – shaped neighbourhood $U' \subset T_p(M)$.

$$\gamma(t) = (tv^1, tv^2, \ldots, tv^m), 0 \le t \le 1 \tag{4.62}$$

and $p = \gamma(0),\ q = \text{Exp}_p(v) = \gamma(1) = (v^1, v^2, \ldots, v^m)$.

(ii) At p, the metric $g = \delta_{ij}$ where $\boldsymbol{\delta} = I_{m\times m}$, the identity matrix.
(iii) Christoffel symbols and the first partial derivatives of g_{ij} vanish at p.

The property (iii) follows from the following arguments.
For all choices of $\boldsymbol{v}$ on T_pM with constant $v^i \neq 0$, the geodesic equation (4.51c) gives:

$$v^i v^j \Gamma_{ij}^k(p) = 0, k = 1, 2, \ldots, n$$

$$\Rightarrow \Gamma_{ij}^k(p) = 0 \tag{4.63}$$

It follows from the last equation that:

$$\left.\frac{\partial g_{ij}}{\partial x_k}\right|_p = 0 \tag{4.64}$$

The last assertion is valid since:

$$\left.\frac{\partial g_{ij}}{\partial x_k}\right|_p = \frac{\partial}{\partial x_k}\left\langle \frac{\partial}{\partial x_i}, \frac{\partial}{\partial x_j}\right\rangle_p = \left\langle \Gamma_{ki}^l \frac{\partial}{\partial x_l}, \frac{\partial}{\partial x_j}\right\rangle_p + \left\langle \frac{\partial}{\partial x_i}, \Gamma_{kj}^l \frac{\partial}{\partial x_l}\right\rangle_p$$

$$= \Gamma_{ki}^l(p) g_{lj}(p) + \Gamma_{kj}^l(p) g_{il}(p)$$

$$= 0 \,(\text{from Equation 4.63}) \tag{4.65}$$

4.2.9 Riemannian Curvature

Curvature is an intrinsic local property of a manifold. If $\gamma(t)$ is a geodesic curve on M, curvature at a point $p \in M$ signifies how much a segment of the curve differs from being a straight line. For the manifold M, it is a measure of how much M differs from being flat at p and M is said to be of zero curvature at p if and only if it is flat at the point. This definition is consistent with our intuitive understanding of curvature in the Euclidean setting. We may define curvature mathematically as follows. With vector fields $X, Y \in TM$, the Riemannian curvature is a map that associates to each pair of vector fields X and Y a differential operator as:

$$R(X,Y) = \nabla_X \nabla_Y - \nabla_Y \nabla_X - \nabla_{[X,Y]} \tag{4.66a}$$

Following the definition above, for a manifold M with a linear connection ∇, the curvature R is the map $R : TM \times TM \times TM \to TM$ given by:

$$R(X,Y)Z = \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X,Y]} Z, X, Y, Z \in TM \tag{4.66b}$$

While torsion in Equation (4.56) refers to non-commutativity of first covariant derivatives, curvature in the last equation may be interpreted as measuring the non-commutativity of the second covariant derivatives. In the case of $M = \mathbb{R}^n$ (where the curvature tensor vanishes), one may observe that for vector fields $X, Y, Z \in \mathbb{R}^n$, $\nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z = \nabla_{[X,Y]} Z$ showing that the operator is indeed commutative. The Riemannian curvature in Equation (4.66) is a linear operator over $C^\infty(M)$, i.e. if $f \in C^\infty(M)$, then:

$$fR(X,Y)Z = R(X,fY)Z = R(fX,Y)Z = R(X,Y)(fZ) \tag{4.67}$$

For instance, let us show that $R(X,Y)(fZ) = fR(X,Y)Z$.

$$R(X,Y)(fZ) = \nabla_X \nabla_Y (fZ) - \nabla_Y \nabla_X (fZ) - \nabla_{[X,Y]} (fZ) \tag{4.68}$$

We expand each term on the RHS:

$$\nabla_X \nabla_Y (fZ) = \nabla_X \left(f \nabla_Y Z + Y(f) Z \right)$$

$$= \left(f \nabla_X \nabla_Y Z + X(f) \nabla_Y Z \right) + \left(Y(f) \nabla_X Z + XY(f) Z \right) \tag{4.69a}$$

(from equation 4.46c)

Similarly,

$$\nabla_Y \nabla_X (fZ) = \nabla_Y \left(f \nabla_X Z + X(f) Z \right)$$

$$= \left(f \nabla_Y \nabla_X Z + Y(f) \nabla_X Z \right) + \left(X(f) \nabla_Y Z + YX(f) Z \right) \tag{4.69b}$$

and

$$\nabla_{[X,Y]} (fZ) = f \nabla_{[X,Y]} Z + \left(XY(f) - YX(f) \right) Z$$

$$\left(\text{from the definition of Lie bracket} \right) \tag{4.69c}$$

Substituting Equation (6.69) in (4.68), we get:

$$R(X,Y)(fZ) = \left(f \nabla_X \nabla_Y Z + XY(f) Z \right) - \left(f \nabla_Y \nabla_X Z + YX(f) Z \right)$$

$$-\left(f\nabla_{[X,Y]}Z+\left(XY\left(f\right)-YX\left(f\right)\right)Z\right)$$

$$=\left(f(\nabla_X\nabla_Y Z-\nabla_Y\nabla_X Z-\nabla_{[X,Y]}Z\right)$$

$$=fR\left(X,Y\right)Z \tag{4.70}$$

Thus, the assertion $R\left(X,Y\right)\left(fZ\right)=fR\left(X,Y\right)Z$ is proved. Similar procedural steps prove the validity of the other equalities in Equation (4.67).

A geometric interpretation of curvature is well known (Lee 1997). For this, let us we consider the parallel transport of a tangent vector on the surface of a sphere S^2. As shown in Figure 4.16, a vector Z parallelly transported along a closed path $N-B-A-N$ fails to come back to the initial orientation at N. The extent of the failure to close by parallel transport around closed loops is called holonomy and curvature is a measure of holonomy (Murray 1996, Christian 2015). It is equivalent to the non-commutativity of the second covariant derivative as defined in Equation 4.66. Further, if ∇ is torsion-free and thus a Levi-Civita connection, the curvature operator R is a linear transformation $T_p\left(M\right)\times T_p\left(M\right)\rightarrow T_p\left(M\right)$. If we consider the vectors X,Y and Z in local coordinates as $X=x^i\dfrac{\partial}{\partial x^i}$, $Y=y^j\dfrac{\partial}{\partial x^j}$ and $Z=z^k\dfrac{\partial}{\partial x^k}$ at p on $T_p\left(M\right)$, then the curvature can be expressed as:

$$R\left(X,Y\right)Z=x^i y^j z^k R^l_{ijk}\frac{\partial}{\partial x^l} \tag{4.71}$$

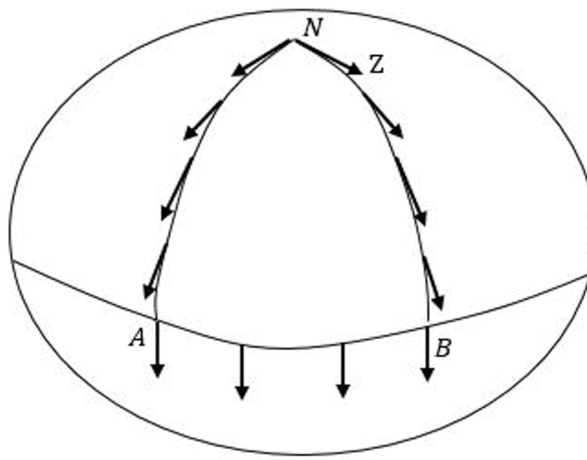


FIGURE 4.16 Riemannian curvature – a measure of holonomy; parallel transport around a closed loop on S^2.

where the coefficients R^l_{ijk} are defined by:

$$R\left(\partial_i, \partial_j\right)\partial_k = R^l_{ijk}\partial_l \tag{4.72}$$

where $\partial_i = \dfrac{\partial}{\partial x^i}, i = 1, 2, \ldots$. From the definition of curvature (Equation 4.66b), the left hand side of the last equation is:

$$\begin{aligned}
R\left(\partial_i, \partial_j\right)\partial_k &= \nabla_{\partial_i}\nabla_{\partial_j}\partial_k - \nabla_{\partial_j}\nabla_{\partial_i}\partial_k - \nabla_{\left[\partial_i, \partial_j\right]}\partial_k \\
&= \left(\nabla_{\partial_i}\nabla_{\partial_j} - \nabla_{\partial_j}\nabla_{\partial_i}\right)\partial_k \left(\text{since } \left[\partial_i, \partial_j\right] = 0\right) \\
&= \left(\nabla_{\partial_i}\Gamma^l_{jk}\partial_l - \nabla_{\partial_j}\Gamma^l_{ik}\partial_l\right)\left(\text{from Equation 4.47} - \nabla_{E_i}E_j = \Gamma^k_{ij}E_k\right) \\
&= \left(\partial_i\left(\Gamma^l_{jk}\right)\partial_l + \Gamma^l_{jk}\nabla_{\partial_i}\partial_l\right) - \left(\partial_j\left(\Gamma^l_{ik}\right)\partial_l + \Gamma^l_{ik}\nabla_{\partial_j}\partial_l\right) \\
&= \left(\frac{\partial\Gamma^l_{kj}}{\partial x^i}\partial_l + \Gamma^l_{jk}\Gamma^m_{il}\partial_m\right) - \left(\frac{\partial\Gamma^l_{ik}}{\partial x^j}\partial_l + \Gamma^l_{ik}\Gamma^m_{jl}\partial_m\right) \\
&= \frac{\partial\Gamma^l_{kj}}{\partial x^i}\partial_l - \frac{\partial\Gamma^l_{ki}}{\partial x^j}\partial_l + \Gamma^l_{jk}\Gamma^m_{il}\partial_m - \Gamma^l_{ik}\Gamma^m_{jl}\partial_m \\
&= \frac{\partial\Gamma^l_{kj}}{\partial x^i}\partial_l - \frac{\partial\Gamma^l_{ki}}{\partial x^j}\partial_l + \Gamma^m_{jk}\Gamma^l_{im}\partial_l - \Gamma^m_{ik}\Gamma^l_{jm}\partial_l \\
&= \left(\frac{\partial\Gamma^l_{kj}}{\partial x^i} - \frac{\partial\Gamma^l_{ki}}{\partial x^j} + \Gamma^m_{jk}\Gamma^l_{im} - \Gamma^m_{ik}\Gamma^l_{jm}\right)\partial_l
\end{aligned} \tag{4.73}$$

From Equations (4.72) and (4.73), we have R^l_{ijk} in Equation (4.71) as:

$$R^l_{ijk} = \frac{\partial\Gamma^l_{kj}}{\partial x^i} - \frac{\partial\Gamma^l_{ki}}{\partial x^j} + \Gamma^m_{jk}\Gamma^l_{im} - \Gamma^m_{ik}\Gamma^l_{jm} \tag{4.74}$$

So far, we have given a brief description of a differentiable manifold M and some of its intrinsic structural properties. The geometric methods of optimization which form the subject of this chapter and rest of the book are based on these properties of manifolds. In the following section, we intend to introduce the basic concepts of

geometric methods. In particular, we first describe how the classical methods – the gradient methods to be specific – may be extended to manifolds.

4.3 GEOMETRIC METHODS OF OPTIMIZATION

A great advantage of a geometric method is to account for any constraint that may be externally specified or naturally arise given the objective function. It is therefore possible that a geometric method might result in faster convergence or higher accuracy. In the exposition of these methods, we presume that the manifold (the characterization of which often depends on the ingenuity of the user) is Riemannian with a unique Levi-Civita connection. In the Riemannian setting, we define an optimization problem as:

$$\text{minimize } f(\boldsymbol{x}),\ \boldsymbol{x} \in M \tag{4.75}$$

where the design variable space is M and $f(.) \in C^{\infty}(M)$. For example, with reference to the minimization problem for the 2D Rosenbrock function (see Figure 4.1), the manifold is represented by the surface $F(\boldsymbol{x}) = (\boldsymbol{x}, f(\boldsymbol{x}))$ with $\boldsymbol{x} = (x_1, x_2)$. Optimization on manifolds may be considered as an unconstrained problem on a surface represented by the manifold M. The objective is to move on the manifold from an initial point $\boldsymbol{x}_0 \in M$, and iteratively reach the optimum $\boldsymbol{x}^* \in M$ through a sequence of updates $\boldsymbol{x}_k \in M, k = 1,2,\ldots$. It is possible to generalize (Gabay 1982, Smith 1994, Absil 2004, 2007a, 2007b) most of the classical methods to optimization on a Riemannian manifold (as stated in Equation 4.75). This is primarily due to the local Euclidean property of the manifold. However, we need to exploit the tools of differential geometry described in the previous section.

Before going into the details of some of these generalizations, let us define the convexity of a scalar-valued function $f(\boldsymbol{x})$ on a manifold. This definition is with respect to a geodesic in lieu of a straight line in the Euclidean case (see Figure 1.6). With a geodesic $\gamma(t)$ connecting two points $p, q \in M$ such that $p = \gamma(0)$ and $q = \gamma(1)$, f is convex over $t \in [0,1]$ if:

$$f(\gamma(t)) \leq tf(p) + (1-t) f(q), \forall t \in [0,1] \tag{4.76}$$

For a Riemannian manifold with Levi-Civita connection, convexity is thus decided by geodesics which are determined by the connection. The latter is dependent on the Riemannian metric. A non-convex problem may be transformed into a convex one by a change of the metric (da Cruz Neto *et al*. 2006, Bento and Melo 2012).

4.3.1 Riemann Geometric Version of Some Classical Gradient Methods

Consider the first-order gradient method – the classical SDM (steepest descent method – Section 2.2.1, Chapter 2). Let us enumerate the algorithmic steps for the manifold (geometric) version of the method (see Table 4.1)

TABLE 4.1
Geometric Descent Method – Details of the Algorithm

Step 1. Start with suitable choice of $\boldsymbol{x}_0$, the initial point on M. Set $k = 1$.

Step 2. Choose a descent gradient direction $\text{grad}\, f(\boldsymbol{x}_k) \in T_{x_k} M$ with $\boldsymbol{x}_k \in M$

Step 3. If $\text{grad}\, f(\boldsymbol{x}_k) \le \epsilon$ (a small number), stop.
Otherwise do a line search and obtain suitable step size $s_k \in R$

Step 4. Set $\boldsymbol{x}_{k+1} = Exp_{x_k}\left(-s_k \,\text{grad}\, f(\boldsymbol{x}_k)\right)$

Step 5. Set $k \equiv k+1$ and go to Step 1.

Steepest descent method
One may find that the key differences between the geometric and classical SDM lie in (i) generating the gradient grad$f(x_k)$ on M in step 2 and (ii) traversing along the geodesic path via the exponential mapping in step 4 with an appropriate step size s_k. The steps involving the generation of grad $f(\boldsymbol{x}_k)$, choice of step size s_k and exponential mapping need some elaboration.
grad $f(x_k)$ *on a manifold* [Lee 1997]

For the objective function $f \in C^\infty(M)$, the gradient of f is the vector field denoted by grad f defined as:

$$\langle \text{grad}\, f, \boldsymbol{v} \rangle_x = df(v),\ \forall \boldsymbol{v} \in T_x M,\ \boldsymbol{x} \in M \tag{4.77a}$$

$df(\boldsymbol{v})$ is the directional derivative (basically a differential or 1-form) of f along $\boldsymbol{v} \in T_X M$ and is equal to $\left\langle \frac{\partial f}{\partial \boldsymbol{x}}, \boldsymbol{v} \right\rangle$ where $\frac{\partial f}{\partial \boldsymbol{x}}$ is the Euclidean gradient. The definition of grad$f$ is consistent with the Euclidean gradient, i.e.:

$$df(\boldsymbol{v}) = \lim_{\varepsilon \to 0} \frac{f(\boldsymbol{x} + \varepsilon v) - f(\boldsymbol{x})}{\varepsilon} = \frac{\partial f}{\partial \boldsymbol{x}}^T \boldsymbol{v} = \left(\frac{\partial f}{\partial \boldsymbol{x}}\right)^i v_i \tag{4.77b}$$

Note that with grad$f$ being a vector field on M, $\langle \text{grad}\, f, \boldsymbol{v} \rangle_x = g_x(\text{grad}\, f, \boldsymbol{v})$. From the definition of a metric on a manifold, one has:

$$\langle \text{grad}f, \boldsymbol{v} \rangle_x = g_{ij} (\text{grad}f)^i v^j \tag{4.77c}$$

From Equations (4.77b–c) and given that the vector $\boldsymbol{v}$ is arbitrary, the Riemannian gradient is obtained as:

$$\text{grad}f(\boldsymbol{x}) = g^{-1} \frac{\partial f}{\partial \boldsymbol{x}} \tag{4.78}$$

As in the Euclidean case, negative of grad$f(\boldsymbol{x})$ is the steepest descent direction.

Step size s_k in steps 2 and 3
The step size may be selected by following any of the classical line search techniques (see Chapter 1).

Exponential mapping in step 4 of Table 4.1, as described in Section 4.2.7 above, maps $-s_k \operatorname{grad} f(\boldsymbol{x}_k)$ on $T_x M$ to the updated point $\boldsymbol{x}_{k+1}$ on the manifold M preserving the length of the tangent vector. However, from a computational point of view, performing the exponential mapping step at each iteration is prohibitively expensive especially for higher dimensional problems. An alternative approach is to approximate this step by what is known as retraction (Absil *et al*. 2007b, Baker and Parks 2016) to get the updates $\boldsymbol{x}_{k+1}$. We discuss this later in this section.

Example 4.5. Consider finding the minimum of Rayleigh quotient $\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x}$ by geometric SDM using the differential geometric approach. Here $\boldsymbol{x} \in R^n$ and $\boldsymbol{A}$ is an $n \times n$ symmetric matrix.

Solution. This is an eigenvalue problem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{\lambda}\boldsymbol{x}$ requiring us to find the lowest eigenvalue and the corresponding eigenvector. Minimizing the Rayleigh quotient (Clough and Penzien 1982) yields the eigenvector $\boldsymbol{x}$ corresponding to the lowest eigenvalue. The eigenvector is an orthonormalized one in that $\boldsymbol{x}^T \boldsymbol{x} = 1$. Here, the problem is solved for $n = 2$. The optimization problem is posed as:

$$\begin{aligned} &\text{minimize } f(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} \\ &\text{s.t. } \boldsymbol{x}^T \boldsymbol{x} = 1 \end{aligned} \tag{4.79}$$

The constraint surface is thus given by $F\left(\boldsymbol{x}, z = \sqrt{1 - \boldsymbol{x}^T \boldsymbol{x}}\right)$. It represents the manifold $M = S^2$, the sphere embedded in R^3. With $x = \boldsymbol{x}(1)$ and $y = \boldsymbol{x}(2)$ and $\boldsymbol{x} = (x, y)^T$, we define local coordinates $u = x,\ v = y$ via a chart (M, ϕ) where ϕ maps an open set $U \supset \boldsymbol{x}$ on M into R^2, the $u - v$ plane. Let ϕ be the coordinate chart with $z = +\sqrt{1 - x^2 - y^2}$ mapping the neighbourhood of any point $p \in M$ on the upper half of the sphere to the $u - v$ plane. Any tangent space $T_p M$ is represented by these coordinates with the metric g given by:

$$\mathfrak{g}_{ij} = \left(\frac{\partial \vec{r}}{\partial u^i} \cdot \frac{\partial \vec{r}}{\partial u^j} \right), i, j = 1, 2 \tag{4.80}$$

With $\vec{r} = \left(x(u,v), y(u,v), z(u,v)\right)$, the metric is given by:

$$\mathfrak{g}_{11} = \left(\begin{Bmatrix} \frac{\partial x}{\partial u} \\ \frac{\partial y}{\partial u} \\ \frac{\partial z}{\partial u} \end{Bmatrix} \cdot \begin{Bmatrix} \frac{\partial x}{\partial u} \\ \frac{\partial y}{\partial u} \\ \frac{\partial z}{\partial u} \end{Bmatrix} \right) = \frac{1 - v^2}{1 - u^2 - v^2}, \mathfrak{g}_{12} = \left(\begin{Bmatrix} \frac{\partial x}{\partial u} \\ \frac{\partial y}{\partial u} \\ \frac{\partial z}{\partial u} \end{Bmatrix} \cdot \begin{Bmatrix} \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial v} \end{Bmatrix} \right) = \frac{uv}{1 - u^2 - v^2} = \mathfrak{g}_{21}$$

$$\mathfrak{g}_{22} = \left(\begin{Bmatrix} \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial v} \end{Bmatrix} . \begin{Bmatrix} \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial v} \end{Bmatrix} \right) = \frac{1-u^2}{1-u^2-v^2} \tag{4.81}$$

With the metric g thus expressed in terms of the local coordinates, we now need to have the steepest descent direction as $-\text{grad} f(\boldsymbol{x}) = -\mathfrak{g}^{-1} \dfrac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$ on $T_x M$. For the objective function $f(\boldsymbol{x})$ on hand, $\dfrac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$ (the Euclidean gradient) is given by:

$$\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = 2\boldsymbol{A}\boldsymbol{x} \tag{4.82}$$

For the matrix $\boldsymbol{A} = \begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix}$, a simple computation yields the lowest eigenvalue as -0.1623 and the corresponding eigenvector as $\boldsymbol{x} = (0.8112, -0.5847)^T$. Figure 4.17 shows the result obtained by minimizing the Rayleigh quotient $\boldsymbol{x}^T\boldsymbol{A}$ by the geometric SDM. The iterative process is started with $\boldsymbol{x}_0 = (0.5, -0.7)$. The path to optimum $\boldsymbol{x}^*$ is

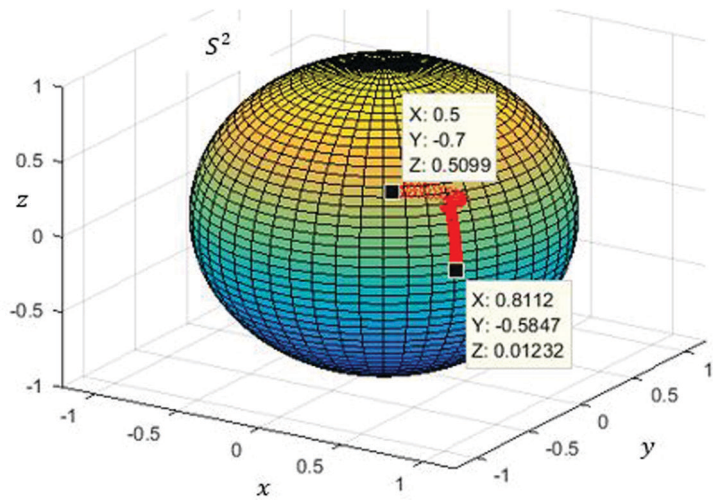


FIGURE 4.17 Riemannian optimization by geometric steepest descent method: minimization of Rayleigh quotient $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$, starting point $\boldsymbol{x}_0 = (0.5, -0.7)^T$ and optimum point $\boldsymbol{x}^* = (0.8112, -0.5847)^T$ with minimum value $f(\boldsymbol{x}^*) = -0.1623$ found in 100 iterations.

plotted in the figure. The minimum value of the objective function is the lowest eigenvalue given by $f\left(\boldsymbol{x}^*\right)$, with $\boldsymbol{x}^*$ being the corresponding eigenvector.

■

The exponential mapping (see Table 4.1) is obviously the vital step in the iteration process. However, it requires solving coupled ODEs (4.51) at each iteration. Retraction is an alternative to the computationally expensive exponential mapping (Absil *et al.* 2007b, Baker and Parks 2016). It approximates the geodesic and provides a first-order approximation to the exponential mapping:

$$R_{x_k}\left(-\text{grad}f\left(\boldsymbol{x}_k\right)\right) \cong Exp_{x_k}\left(-\text{grad}f\left(\boldsymbol{x}_k\right)\right) = \boldsymbol{x}_k - \text{grad}f\left(\boldsymbol{x}_k\right) \tag{4.83}$$

It is possible to have different retractions (Ring and Wirth 2012) to relax the exponential mapping. Figure 4.18 shows the result for the example problem 4.5 by replacing the exponential mapping step 4 of Table 4.1 by retraction.

Conjugate gradient method (CGM)

As in the classical CGM, the geometric CG method differs from SDM in generating the new direction and hence the update $\boldsymbol{x}_{k+1}$ in step 4 of Table 4.1. The other steps involving gradient generation and exponential mapping remain the same. Also, the method adopts the same strategy as in the classical CGM in obtaining the new direction at each iteration. That is, the new direction $\boldsymbol{d}_k$ is not just $-\text{grad}f\left(\boldsymbol{x}_k\right)$, but is so

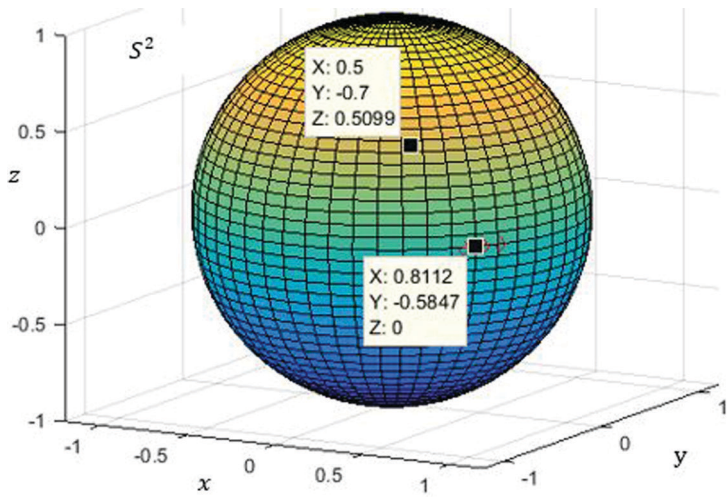


FIGURE 4.18 Use of retraction in Riemannian optimization, result by geometric steepest descent method, minimization of Rayleigh quotient $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$, starting point $\boldsymbol{x}_0 = \left(0.5,-0.7\right)^T$ and optimum point $\boldsymbol{x}^* = \left(0.8112,-0.5847\right)^T$ with minimum value $f\left(\boldsymbol{x}^*\right) = -0.1623$ found in 24 iterations.

constructed that it is conjugate to the direction $\boldsymbol{d}_{k-1} \in T_{x_{k-1}} M$ of the previous step as in the classical CGM (Equations 2.30 and 2.36 in Chapter 2). But it differs in implementation. Specifically, we note that the gradient vectors $\boldsymbol{d}_{k-1}$ and $\text{grad}f\left(\boldsymbol{x}_k\right)$ lie in different tangent spaces. Before any vector operation could be performed on these two vectors, we first need to do a parallel transport of $\boldsymbol{d}_{k-1}$ to the current point $\boldsymbol{x}_k \in M$. If $\overline{\boldsymbol{d}}_{k-1}$ is the vector corresponding to the parallel transport of $\boldsymbol{d}_{k-1}$ to $\boldsymbol{x}_k$, the new search direction at $\boldsymbol{x}_k$ is obtained as:

$$\boldsymbol{d}_k = -\text{grad}f\left(\boldsymbol{x}_k\right) + \beta_{k-1}\overline{\boldsymbol{d}}_{k-1} \tag{4.84}$$

$\beta_{k-1} \in R$ is a parameter introduced as in the classical CGM method (Equation 2.36, Section 2.2.2, Chapter 2) and similar to Fletcher and Reeves (1964). It is given by:

$$\beta_{k-1} = \frac{\text{grad}f\left(\boldsymbol{x}_k\right)^T \text{grad}f\left(\boldsymbol{x}_k\right)}{\overline{\text{grad}}f\left(\boldsymbol{x}_{k-1}\right)^T \overline{\text{grad}}f\left(\boldsymbol{x}_{k-1}\right)} \tag{4.85}$$

$\overline{\text{grad}}f\left(\boldsymbol{x}_{k-1}\right)$ corresponds to the parallel transport of the Riemannian gradient $\text{grad}f\left(\boldsymbol{x}_{k-1}\right)$ to current point $\boldsymbol{x}_k \in M$. A discussion on other possibilities for the parameter β_{k-1} may be found in Smith (1993) and Boumal (2014). The following example illustrates the application of geometric CGM for the Rosenbrock function.

Example 4.6. Find the minimum of the Rosenbrock function $f(x_1, x_2) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2$ by geometric CGM.

Solution. The Rosenbrock function is in fact solved by classical methods for the minimum in Chapter 2. Here, we obtain the solution by the Riemannian version of CGM with the convergence criterion of keeping $\varepsilon = \left(f\left(\boldsymbol{x}_k\right) - f\left(\boldsymbol{x}_{k+1}\right)\right)$ within a tolerance of 1e-10 using the MANOPT software (Boumal *et al.* 2014). Solution in terms of the optimum path and evolution of the objective function is shown in Figures 4.19a–b. Result by geometric SDM is also included in the figures (see Figures 4.19c–d). CGM has shown better result compared to SDM as expected. Since it is a two-dimensional problem, the matrix $\mathfrak{g}$ for the Riemannian metric is obtained by embedding the manifold surface in $\mathbb{R}^3$ as $F\left(x_1, x_2, z = f(x_1, x_2)\right)$ and by following the procedure similar to the one in Example 4.5. Expressions for the elements of $\mathfrak{g}$ in terms of local coordinates x_1 and x_2 are given in Appendix 4 (item A4.2). Resulting evolutions of the objective function by classical CGM and SDM are included in Figures 4.19b and 4.19d, respectively. Convergence is no doubt achieved by the classical CGM (see Figure 4.19b) but at a slower rate vis-à-vis the geometric CGM. By the classical SDM, convergence is not realized at all and an oscillatory behaviour is noticed even after 2000 iterations (see the dotted line in Figure 4.19d).

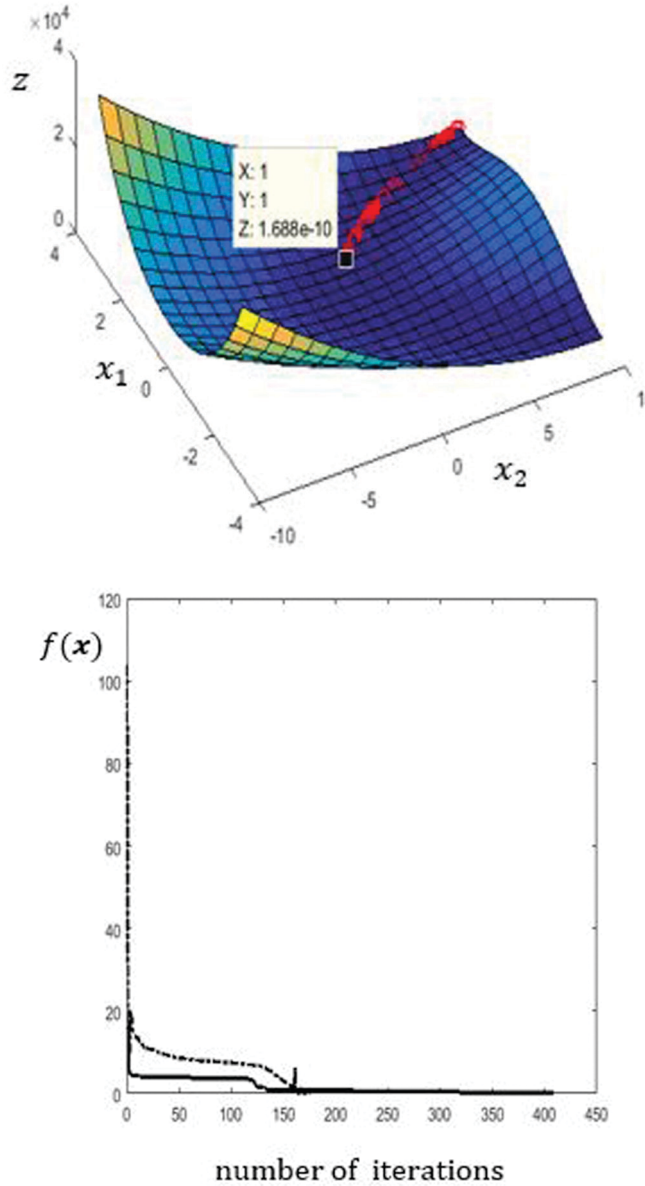


FIGURE 4.19a–b Optimization by geometric conjugate gradient method – Rosenbrock function: (a) optimum path to $\boldsymbol{x}^*$ on the manifold and (b) evolution of the objective function with iterations, dark line – geometric CGM and dash-dotted line – classical CGM. Optimization by geometric steepest descent method – Rosenbrock function: (c) optimum path to $\boldsymbol{x}^*$ on the manifold and (d) evolution of the objective function with iterations with log scale on y-axis; dark line – geometric SDM and dotted line – classical SDM (oscillatory behaviour and no convergence). Search paths: (e) classical SDM and (f) classical CGM; note the zig-zag paths following line search at each iteration which increases the computational effort.

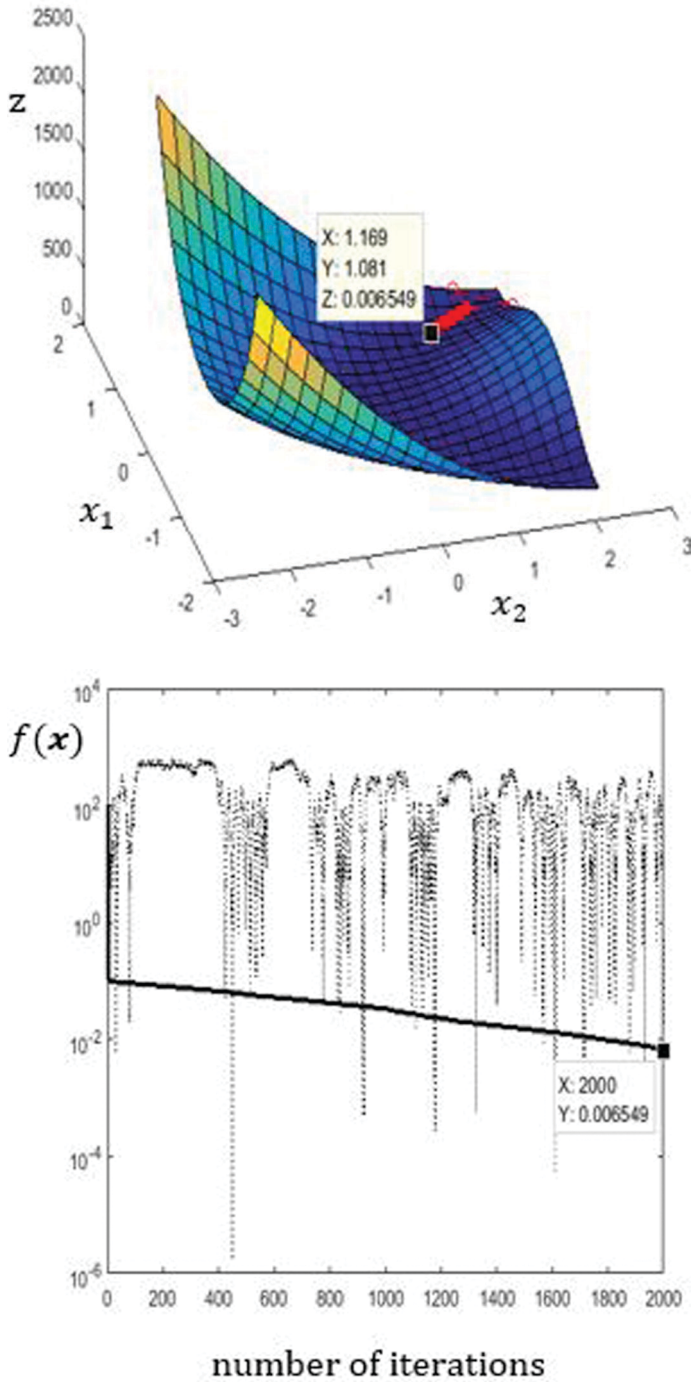


FIGURE 4.19c–d (Continued)

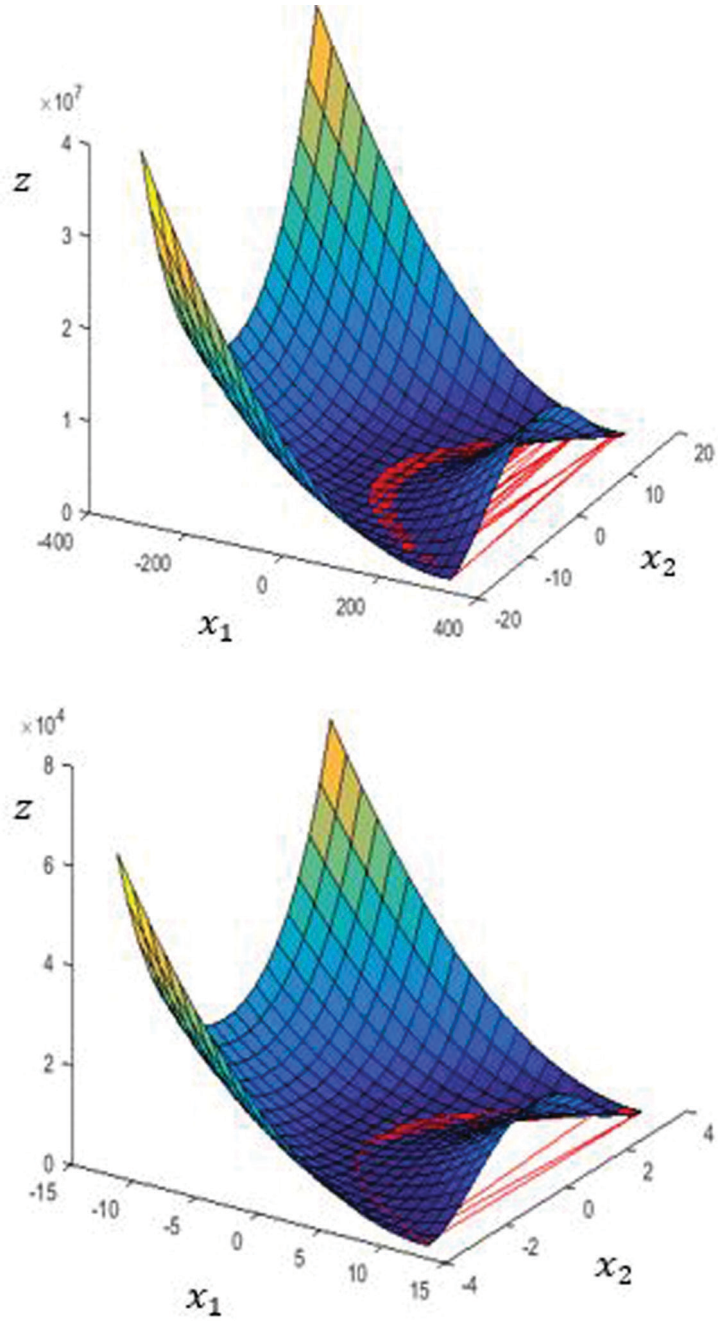


FIGURE 4.19e–f (Continued)

The search paths during the optimization process by classical SDM and CGM are shown in Figures 4.19e–f. As expected, the geometric versions of the two methods offers a substantively smoother evolution for the search path before finally reaching $\boldsymbol{x}^*$. The geometric SDM has linear convergence and the CG version has quadratic convergence (Smith 1994).

Newton's method (NM)

The geometric version of Newton's method requires that the Hessian matrix be appropriately defined on the manifold M.With $\text{grad } f(\boldsymbol{x}) \in T_x M$ as the Riemannian gradient (Equation 4.78) of a function $f \in C^\infty(M)$, the Hessian of f at $p \in M$ with connection ∇ is the covariant derivative of $\text{grad } f$ (which is identifiable with the 1-form df) and is given by:

$$\text{Hess}(f) := \nabla\nabla f = \nabla^2 f : T_p M \times T_p M \to \mathbb{R} \tag{4.86}$$

Given two vector fields $X, Y \in TM$, $\text{Hess}(f)(X,Y) := \nabla^2 f(X,Y)$ is defined by (Hsu 2002):

$$\nabla^2 f(X,Y) = X(Yf) - (\nabla_X Y) f \tag{4.87}$$

In terms of local coordinates, the two terms on the RHS of the last equation take the from:

$$X(Yf) = X^i E_i (Yf) = X^i E_i \left(Y^j E_j f\right) = X^i \frac{\partial Y^j}{\partial x^i}\frac{\partial f}{\partial x^j} + X^i Y^j \frac{\partial E^j}{\partial x^i} f + X^i Y^j \frac{\partial^2 f}{\partial x^i \partial x^j}$$

$$= X^i \frac{\partial Y^j}{\partial x^i}\frac{\partial f}{\partial x^j} + X^i Y^j \frac{\partial^2 f}{\partial x^i \partial x^j} \left(\text{since} \frac{\partial E^j}{\partial x^i} = 0\right) \tag{4.88a}$$

and

$$(\nabla_X Y) f = \left(\nabla_X \left(Y^j E_j\right)\right) f = \left(X^i \frac{\partial Y^j}{\partial x^i} E_j + X^i Y^j \nabla_{E_i} E_j\right) f$$

$$= X^i \frac{\partial Y^j}{\partial x^i}\frac{\partial f}{\partial x^j} + X^i Y^j \Gamma_{ij}^k E_k f = X^i \frac{\partial Y^j}{\partial x^i}\frac{\partial f}{\partial x^j} + X^i Y^j \Gamma_{ij}^k \frac{\partial f}{\partial x^k} \tag{4.88b}$$

Therefore, the Hessian operator is obtained from Equation (4.87) as:

$$\text{Hess}(f) = \nabla^2 f = \left(\frac{\partial^2 f}{\partial x^i \partial x^j} - \Gamma_{ij}^k \frac{\partial f}{\partial x^k}\right) \tag{4.89}$$

$\frac{\partial^2 f}{\partial x^i \partial x^j}$ is the Euclidean Hessian of the objective function $f(\boldsymbol{x})$. The symmetric property of the Riemannian connection (with zero torsion) renders $\text{Hess}\, f(\boldsymbol{x})$ a self-adjoint symmetric bi-linear operator with respect to the matric g:

$$\left\langle \text{Hess}\, f(\boldsymbol{x})[\boldsymbol{v}], \boldsymbol{w} \right\rangle = \left\langle \text{Hess}\, f(x)[\boldsymbol{w}], \boldsymbol{v} \right\rangle, \boldsymbol{v}, \boldsymbol{w} \in T_x M \tag{4.90}$$

Update in Newton's method

Using the Hessian on the manifold as defined above, the update in Newton's method is obtained as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \left(\text{Hess} f(\boldsymbol{x})\right)^{-1} \text{grad} f(\boldsymbol{x}_k) \tag{4.91}$$

Note that there is no need of parallel transport in the geometric NM unlike in the CG. But in developing the Riemannian versions of quasi-Newton methods (Section 2.3, Chapter 2), parallel transport operation is required in getting the update. For details on these versions, see Gabay (1982), Smith (1993), and Huang (2013).

Trust region method (TRM)

In the Riemannian setting, the trust region method also closely follows the Euclidean version (Section 3.3.3, Chapter 3) exhibiting similar convergence properties (Baker 2008). At each iteration, a quadratic model (Equation 3.30, Chapter 3) $q(\boldsymbol{x})$ approximating the objective function $f(\boldsymbol{x})$ is solved within a pre-specified trust region. The trust region at the current $\boldsymbol{x}_k$ is treated as a manifold endowed with a metric. The quadratic model within this region is defined as (Absil *et al*. 2007a):

$$q(\boldsymbol{x}_k) = f(\boldsymbol{x}_k) + \left\langle (\boldsymbol{x} - \boldsymbol{x}_k), \text{grad}\, f(\boldsymbol{x}_k) \right\rangle + \frac{1}{2} \left\langle (\boldsymbol{x} - \boldsymbol{x}_k), \text{Hess}\, f(\boldsymbol{x} - \boldsymbol{x}_k) \right\rangle \tag{4.92}$$

The quadratic model is trusted within a ball of radius Δ_k and may be solved by, say Newton's method in the Riemannian setting. The trust region radius is updated depending upon a parameter R_k (Equation 3.35, Chapter 3) which is the ratio of the actual reduction in $f(\boldsymbol{x}_k)$ to the one in $q(\boldsymbol{x}_k)$ similar to the classical case.

Example 4.7. We again consider the Rosenbrock function $f(\boldsymbol{x})$ = 100 $\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2$ of Example 4.6 and solve it by geometric NM and TRM.

Solution. The results from the two methods are shown in Figures 4.20 and 4.21. Since the methods are second order, the convergence is faster compared to SDM and CGM.

Results by the classical Newton and trust region methods are also shown respectively in Figures 4.20b and 4.21b (dash-dotted lines) wherein the evolution of the objective function

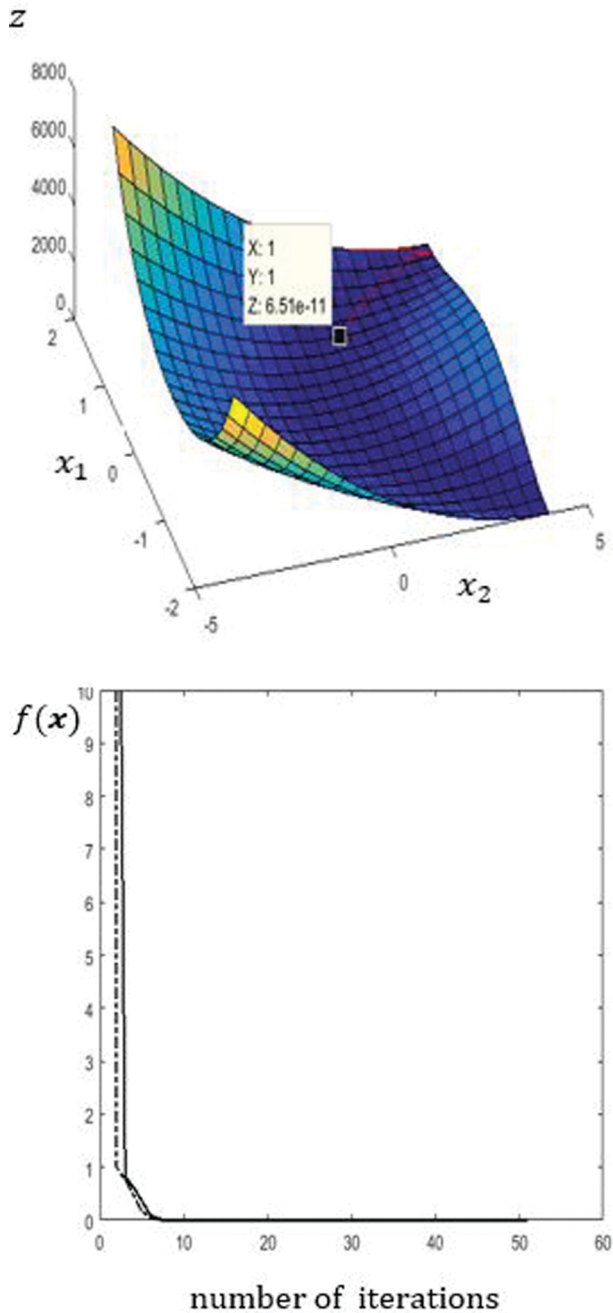


FIGURE 4.20 Optimization by geometric Newton's method (NM) – Rosenbrock function: (a) optimum path to x^* on the manifold and (b) evolution of the objective function with iterations. Dark line – geometric NM and dash-dotted line – classical NM.

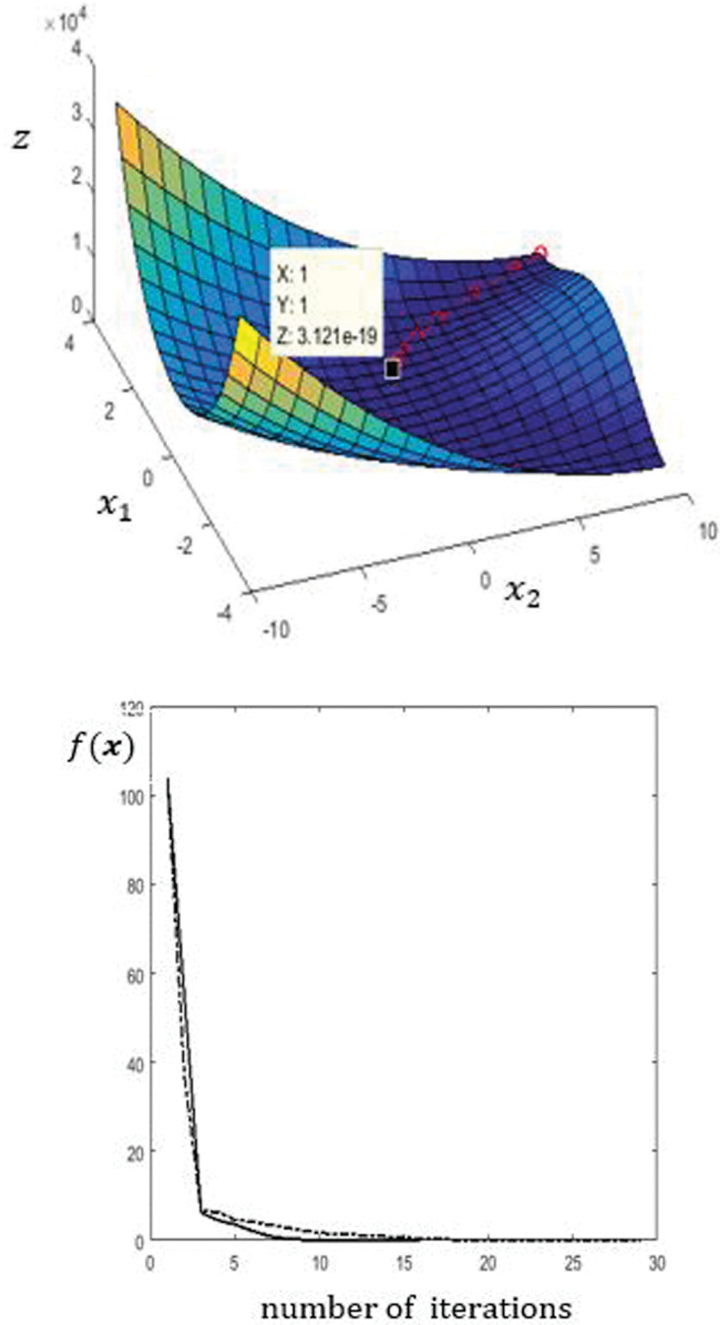


FIGURE 4.21 Geometric optimization by trust region method (TRM) – Rosenbrock function: (a) optimum path to $\boldsymbol{x}^*$ on the manifold and (b) evolution of the objective function with iterations, dark line – geometric TRM and dash-dotted line – classical TRM.

indicates almost the same trend as the corresponding geometric versions. This implies that the geometric versions preserve (but do not improve upon) the quadratic convergence of the two classical schemes.

■

Some possible improvements to the Riemannian versions of the above methods are available in the literature. For example, readers may see Baker (2008), Sato (2013) for details. Studies on these geometric methods include applications to large-scale eigenvalue problems (Absil *et al*. 2007b), signal processing (Smith 2005, Hegde 2012, Manton 2013), data mining (Ma and Fu 2011) and machine learning (Cayton 2005). One may also find Riemannian versions for derivative-free methods (Chapter 3) along with their convergence properties. For this class of problems, interested readers may refer to Dreisigmeyer (2007) and Fong and Tino (2019).

4.4 STATISTICAL ESTIMATION BY GEOMETRICAL METHOD OF OPTIMIZATION

In the introduction to this chapter, a mention is made on the importance of statistical estimation/inference problems (Amari 1983) involving optimization on manifolds. Important application areas are data analysis (Cowan 1998), filtering (Jazwinski 1970, Alspach and Sorenson 1972) and neural networks (Amari 1997, Du and Swamy 2014). In this context, we start with a basic description of statistical estimation using Riemannian geometry.

We have presented an estimation problem – the MLE problem in Section 3.2.2, Chapter 3, where the parameters of a probability distribution were estimated to optimally fit an observed data by using a classical derivative-free method. As elucidated therein, suppose that the observed data is $z = \left\{z_1, z_2, \ldots, z_n\right\}^T$. The observations are assumed to be realizations of random variables (RVs) $Z_i, i = 1,2,\ldots,n$ which should follow the given probability distribution $\mathbb{F}_{\boldsymbol{Z}}\left(z;\boldsymbol{\theta}\right)$, of course provided that the parameters are appropriately estimated. Here $\boldsymbol{Z}$ is the vector of the RVs $Z_i, i = 1,2,\ldots,n$. The vector $\boldsymbol{\theta} \in R^m$ comprises the unknown parameters in the *pdf* $\mathbb{f}_{\boldsymbol{Z}}(.)$ and are to be estimated. Note that, within the MLE scheme, the parameters $\theta_1, \theta_2, \ldots, \theta_m$ are themselves considered as RVs. This is so since, by having different realizations of z, one may have different estimates by the MLE and obtain an approximate sampling distribution of $\hat{\boldsymbol{\theta}}$. This would afford additional information on their respective confidence intervals. See Section 3.2.2, Chapter 3, for more details on this aspect. In the MLE, we get an estimate of $\boldsymbol{\theta}$ via a minimization of the negative log-likelihood function $l\left(\boldsymbol{\theta};\boldsymbol{Z}\right)$. See Equation (3.14b) for $l\left(\boldsymbol{\theta};\boldsymbol{Z}\right)$ which is re-written below:

$$l(\boldsymbol{\theta};\boldsymbol{Z}) = \sum_{i=1}^{n} \log \mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta}) \tag{4.93}$$

$l\left(\boldsymbol{\theta};\boldsymbol{Z}\right)$ is known as the loss function[††] in information theory. Here log stands for the natural logarithm. It is shown in Chapter 3 that for large *n*, the MLE yields an estimate

[††] loss function
A loss function is also error function signifying the error (loss) due to model prediction. In the MLE, the log likelihood function is the loss function and it is an indicator of the prediction error with respect to the available samples in the data set.

close to the true value with an accuracy related to the Fisher information matrix (FIM) defined by:

$$\begin{aligned}\boldsymbol{I}(\boldsymbol{\theta}) &= E_Z\left[\left(\frac{\partial l(\boldsymbol{\theta};\boldsymbol{Z})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial l(\boldsymbol{\theta};\boldsymbol{Z})}{\partial \boldsymbol{\theta}}\right)^T\right] = -E_{\boldsymbol{Z}}\left[\frac{\partial^2 l(\boldsymbol{\theta};\boldsymbol{Z})}{\partial^2 \boldsymbol{\theta}}\right] \\ &= -E_{\boldsymbol{Z}}\left[\boldsymbol{H}(\boldsymbol{\theta})\right]\end{aligned} \tag{4.94}$$

$E_Z[.]$ is the expectation operator with respect to the probability measure corresponding to $\boldsymbol{Z}$: $d\mathbb{F}_{\boldsymbol{Z}} = \mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})$. The differential operators $\frac{\partial}{\partial \boldsymbol{\theta}} = \left(\frac{\partial}{\partial \theta_1}, \frac{\partial}{\partial \theta_2}, \cdots \frac{\partial}{\partial \theta_m}\right)^T$ and $\frac{\partial^2}{\partial \boldsymbol{\theta}^2} = \left[\frac{\partial^2}{\partial \theta_i \partial \theta_j}, i, j = 1, 2, \ldots, m\right]_{m \times m}$. Now, we refer to the Kullback–Leibler (KL) distance or divergence (Kullback and Leibler 1951) between two probability distributions $P(x)$ and $Q(x)$ defined by:

$$\mathcal{D}\left(P(x) \| Q(x)\right) := \int_{x \in \mathbb{C}} \log \frac{P(x)}{Q(x)} p(x) dx = E_P\left[\log \frac{P(x)}{Q(x)}\right] \tag{4.95}$$

KL distance is also known as relative entropy.[‡‡] In the present context, we may have an analogous definition for the KL divergence with respect to the *pdf*s corresponding to $\boldsymbol{\theta}$ and $\boldsymbol{\theta} + \Delta\boldsymbol{\theta}$ in terms of the loss function $l(\boldsymbol{\theta};\boldsymbol{Z})$ as:

$$\begin{aligned}\mathcal{D}\left(\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}+\Delta\boldsymbol{\theta}) || \mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})\right) &= E_{\boldsymbol{z}}\left[\log \frac{\Pi_i \mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta}+\Delta\boldsymbol{\theta})}{\Pi_i \mathbb{f}_{Z_i}(z_i;\boldsymbol{\theta})}\right] \\ &= E_{\boldsymbol{Z}}\left[l(\boldsymbol{\theta}+\Delta\boldsymbol{\theta};\boldsymbol{Z}) - l(\boldsymbol{\theta};\boldsymbol{Z})\right] \\ &= E_{\boldsymbol{Z}}\left[l(\boldsymbol{\theta}+\Delta\boldsymbol{\theta};\boldsymbol{Z}) - l(\boldsymbol{\theta}+\Delta\boldsymbol{\theta}-\Delta\boldsymbol{\theta};\boldsymbol{Z})\right]\end{aligned} \tag{4.96a}$$

In general, the KL divergence $\mathcal{D}\left(P(x) \| Q(x)\right)$ between two probability measures $P(x)$ and $Q(x)$ is defined only if Q is absolutely continuous with respect to $P(x)$, i.e. if, for all $x, Q(x) = 0$ implies $P(x) = 0$. This condition is satisfied for $\mathbb{f}_{\theta}$ and $\mathbb{f}_{\theta+\Delta\theta}$. With $\boldsymbol{\theta}$ as the true value for the MLE and by truncated Taylor's expansion of the second term on the RHS of the last equation about $\boldsymbol{\theta} + \Delta\boldsymbol{\theta}$, one has:

---

‡‡ relative entropy
This is also known as cross entropy; it is a measure of divergence between two probability distributions. It is equivalent to KL divergence within the context of information theory (Gray 1990) for measuring similarity between two *pdf*s. It is widely used in machine learning optimization tasks (Abdolmaleki 2015).

$$\mathcal{D}\left(\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}+\Delta\boldsymbol{\theta})||\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})\right) \cong E_{\boldsymbol{Z}}\Bigg[l(\boldsymbol{\theta}+\Delta\boldsymbol{\theta};\boldsymbol{Z}) - \begin{Bmatrix} l(\boldsymbol{\theta}+\Delta\boldsymbol{\theta};\boldsymbol{Z}) - \Delta\boldsymbol{\theta}l'(\boldsymbol{\theta}+\Delta\boldsymbol{\theta};\boldsymbol{Z}) \\ +\frac{1}{2}\Delta\boldsymbol{\theta}^T l''(\boldsymbol{\theta}+\Delta\boldsymbol{\theta};\boldsymbol{Z})\Delta\boldsymbol{\theta} \end{Bmatrix}\Bigg]$$

$$= E_Z\left[\Delta\theta l'\left(\theta+\Delta\theta;Z\right)-\frac{1}{2}\Delta\boldsymbol{\theta}^T l''\left(\boldsymbol{\theta}+\Delta\boldsymbol{\theta};\boldsymbol{Z}\right)\Delta\boldsymbol{\theta}\right] \tag{4.96b}$$

Noting that $E_Z\left[l'\left(\boldsymbol{\theta}+\Delta\boldsymbol{\theta};\boldsymbol{Z}\right)\right]_{\Delta\theta=0}=0$ (Equation 3.17, Section 3.2.2, Chapter 3) and we have:

$$\mathcal{D}\left(\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}')||\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})\right)=\frac{1}{2}\Delta\boldsymbol{\theta}^T E_{\boldsymbol{Z}}[-l''(\boldsymbol{\theta}+\Delta\boldsymbol{\theta};\boldsymbol{Z})]\Delta\boldsymbol{\theta}$$

$$\approx \Delta\boldsymbol{\theta}^T\boldsymbol{I}\left(\boldsymbol{\theta}\right)\Delta\boldsymbol{\theta}\left(\text{from Equation 3.22}\right) \tag{4.97}$$

Here, with regard to the last equation, we make two observations.

(1) With $\boldsymbol{I}\left(\boldsymbol{\theta}\right)$ being the FIM, the KL divergence is thus an inner product in terms of the FIM.
(2) Also, since MLE aims at minimization of the negative log likelihood function, $\boldsymbol{I}\left(\boldsymbol{\theta}\right)$ should be positive definite at the minimum point (if it exists).

These two observations make it possible to solve the MLE problem by posing it in a Riemannian setting. To this end, we consider a manifold *M* of the same dimension *m* (as the Euclidean case) representing a statistical model that comprises of a set of joint *pdf*s $\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\theta)$. Thus, each point, say, $\boldsymbol{x}$ on *M* has the coordinates denoted by the *m*-dimensional parameter $\boldsymbol{\theta}=\left(\theta^1,\theta^2,\ldots,\theta^m\right)$. Note that consistent with the notation used for coordinates of a point on a manifold, the coordinate elements of $\boldsymbol{\theta}$ are written with superscripts. Also, we unbold θ hereafter.

We now wish to perform the minimization of the negative log likelihood function on the *m*-dimensional, intrinsically curved *M* by geodesic search in lieu of a line search adopted in $\mathbb{R}^m$. If, for the sake of brevity, we denote the *pdf* $\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})$ by $\mathbb{f}_{\boldsymbol{\theta}}$, then $\mathbb{f}_{\boldsymbol{\theta}+\Delta\boldsymbol{\theta}}$ represents a possible joint *pdf* at $\theta+\Delta\theta$ in the neighbourhood of θ (see Figure 4.22).

With $l\left(\theta;\boldsymbol{Z}\right)$ considered as a smooth function on *M*, the KL distance $\mathcal{D}\left(\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}+\Delta\boldsymbol{\theta})||\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})\right)$ signifies how $\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta}+\Delta\boldsymbol{\theta})$ differs from $\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})$ in an averaged sense and is suggestive of a sort of 'distance' between $\boldsymbol{x}$ and $\boldsymbol{y}$ on the manifold *M*. In this scenario, the inner product in Equation (4.97) may be taken as $\left\langle\Delta\boldsymbol{\theta},\Delta\boldsymbol{\theta}\right\rangle_x=\left(\Delta\boldsymbol{\theta}^T\,\mathfrak{g}\Delta\boldsymbol{\theta}\right)_x$ with $\Delta\boldsymbol{\theta}\in T_xM$ and with the metric $\mathfrak{g}$ given by the FIM. Hence, *M* is Riemannian with a metric $\mathfrak{g}$. With these considerations, we illustrate solving the MLE by the geometric version of SDM.

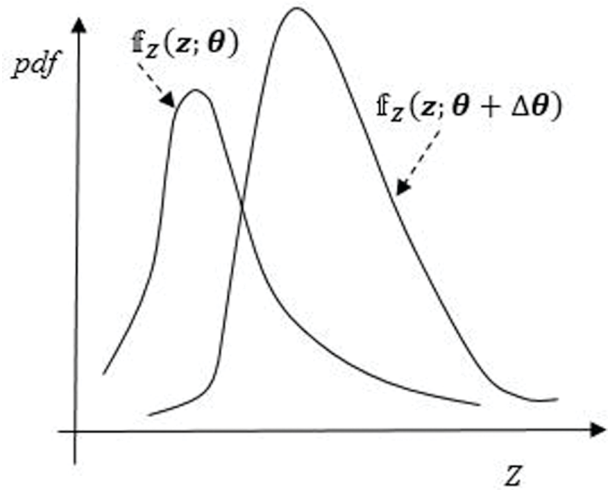


FIGURE 4.22 Joint *pdf*s $\mathbb{f}_Z(\boldsymbol{z};\boldsymbol{\theta})$ and $\mathbb{f}_Z(\boldsymbol{z};\boldsymbol{\theta}+\Delta\boldsymbol{\theta})$ corresponding to points $\boldsymbol{x}$ and $\boldsymbol{y}$ on the manifold M representing an m-dimensional parameter space.

Example 4.8. We consider the statistical estimation problem in Example 3.2 in Chapter 3 and solve the MLE by the geometric SDM.

Solution. In this problem, $\mathbb{f}_Z(z_i;\theta)$ is the generalized exponential *pdf* given by:

$$\mathbb{f}_Z(z;\boldsymbol{\theta}) = \alpha\lambda e^{-\lambda z}(1-e^{-\lambda z})^{\alpha-1},\ z>0 \tag{4.98}$$

$\boldsymbol{\theta}=(\alpha,\lambda)^T$. Let $n=5000$ which corresponds to the number of observations $z_i, i=1,2,\ldots,n$. These observations are in fact obtained by MC simulation of the *pdf* $\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})$ using, say, the inversion method (Appendix 3) with the true values $\alpha=3.639$ and $\lambda=2.239$. The objective is to estimate these two parameters α and λ by MLE using the geometric SDM. This problem is two-dimensional with $m=2$. Following the steps in Table 4.1 of SDM, we first form the Riemannian gradient required in step 2. With $\boldsymbol{\theta}_k \in M$ and the metric $\mathfrak{g}=-\boldsymbol{I}(\boldsymbol{\theta}_k)$ at the k^{th} iteration, Equation (4.78) gives grad $l(\boldsymbol{\theta}_k;\boldsymbol{Z}) \in T_{\boldsymbol{\theta}_k}M$ as:

$$\text{grad}\, l(\boldsymbol{\theta}_k;\boldsymbol{Z}) = -\left[\boldsymbol{I}(\boldsymbol{\theta}_k)\right]^{-1}\frac{\partial l}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}_k;\boldsymbol{Z}) \tag{4.99}$$

where $\dfrac{\partial l}{\partial \boldsymbol{\theta}}$ is the Euclidean gradient:

$$\frac{\partial l}{\partial \boldsymbol{\theta}}(\theta_k;\boldsymbol{Z}) = \frac{\partial}{\partial\theta}\sum_{i=1}^{n}\log \mathbb{f}_Z(z;\theta)\Big|_{\boldsymbol{\theta}_k} \text{ with}$$

$$\log \mathbb{f}_Z(z;\theta) = \log\alpha_k + \log\lambda_k - \lambda_k z + (\alpha_k-1)\log\left(1-e^{-\lambda_k z}\right) \tag{4.100}$$

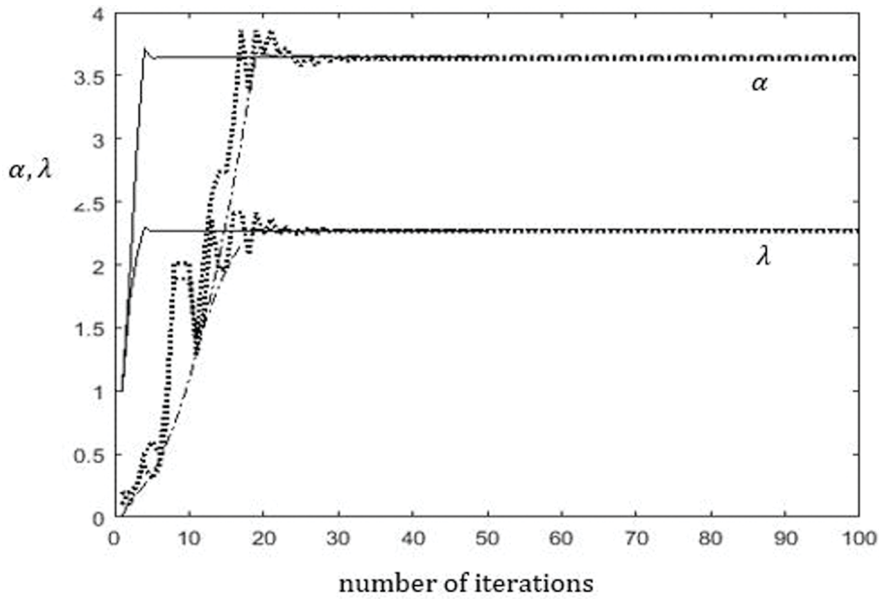


FIGURE 4.23 Statistical estimation by MLE of parameters of generalized exponential probability distribution; evolution of parameters α and λ with iterations; dark line – geometric SDM method, dotted line – classical derivative-free NM method, dash-dotted line – classical derivative-free HJ method.

Therefore, at the k^{th} iteration, we obtain:

$$\frac{\partial l}{\partial \boldsymbol{\theta}}\left(\boldsymbol{\theta}_k;\boldsymbol{Z}\right)=\begin{pmatrix} \dfrac{n}{\alpha_k}+\displaystyle\sum_{i=1}^{n}\log\left(1-e^{-\lambda_k z_i}\right) \\ \dfrac{n}{\lambda_k}-\displaystyle\sum_{i=1}^{n}z_i+\left(\alpha_k-1\right)\sum_{i=1}^{n}\dfrac{z_i e^{-\lambda_k z_i}}{1-e^{-\lambda_k z_i}} \end{pmatrix} \tag{4.101}$$

The FIM $\boldsymbol{I}_n\left(\boldsymbol{\theta}\right)$ is evaluated using Equation (4.94), i.e.:

$$\boldsymbol{I}\left(\boldsymbol{\theta}_k\right)=E_Z\left[\left.\frac{\partial^2 l\left(\boldsymbol{\theta};\boldsymbol{Z}\right)}{\partial^2\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}_k}\right]$$

$$=\begin{bmatrix} -\dfrac{n}{{\alpha_k}^2} & \displaystyle\sum_{i=1}^{n}\dfrac{z_i e^{-\lambda_k z_i}}{1-e^{-\lambda_k z_i}} \\ \displaystyle\sum_{i=1}^{n}\dfrac{z_i e^{-\lambda_k z_i}}{1-e^{-\lambda_k z_i}} & -\dfrac{n}{\lambda_k^2}-\left(\alpha_k-1\right)\displaystyle\sum_{i=1}^{n}\dfrac{z_i^2 e^{-\lambda_k z_i}}{\left(1-e^{-\lambda_k z_i}\right)^2} \end{bmatrix} \tag{4.102}$$

With $\frac{\partial l}{\partial \boldsymbol{\theta}}\left(\boldsymbol{\theta}_k; \boldsymbol{Z}\right)$ and $\boldsymbol{I}\left(\boldsymbol{\theta}_k\right)$ computed at each iteration according to Equations (4.101–102), the Riemannian gradient is evaluated from Equation 4.99. Following Table 4.1, results are obtained by the geometric SDM for the estimated values of parameters α and λ; these are shown in Figure 4.23. Results from the classical derivative-free HJ (Hooke and Jeeves) and NM (Nelder and Mead) methods (Sections 3.2.1 and 3.2.2, Chapter 3) are also included in the same figure. The optimum point θ^* obtained is $(3.639, 2.268)^T$ by the geometric SDM, $\left(3.629, 2.268\right)^T$ by the classical NM method and $\left(3.660, 2.280\right)^T$ by the classical HJ method. By the geometric SDM, θ^* is reached with higher accuracy and at a faster rate vis-à-vis classical derivative-free NM and HJ methods.

■

4.5 ANALOGY BETWEEN STATISTICAL SAMPLING AND STOCHASTIC OPTIMIZATION

As noted in the introduction to this chapter, the underlying concept in the development of optimization methods based on Langevin dynamics is the analogy that exists between statistical sampling and stochastic optimization (Dalalyan 2017a,b). In statistical sampling, a target *pdf* is given and it is required to generate samples (realizations) of the RV associated with the *pdf*. In these sampling methods based on MCMC strategy (Appendix 3), one usually starts with a proposal (assumed) *pdf* and generates samples in the form of a discrete Markov chain whose limiting distribution is expected to match with the target *pdf*. The task may be more elegantly accomplished by solving a Langevin SDE and exploiting its possible convergence to a stationary *pdf* (Smith and Roberts 1993, Welling and Teh 2011, Durmus *et al*. 2019). If a Markov chain is irreducible and aperiodic (Appendix 3), it has a unique stationary distribution which is the limiting distribution. With the property of irreducibility and aperiodicity, a Markov chain is known to be ergodic. Ergodic Markov chains reach the limiting distribution regardless of the initial probabilities of the states.

4.5.1 Langevin SDE – Convergence to a Stationary *pdf*

Suppose we aim at drawing samples from a given target *pdf*. For simplicity of exposition, let us consider the one-dimensional Langevin equation (4.3b) with inertia term ignored as described by the SDE:

$$dX_t = \alpha\left(X_t\right)dt + \sigma\left(X_t\right)dB\left(t\right), \qquad X\left(0\right) = X_0 \tag{4.103}$$

Equation (4.103) is also known as the overdamped Langevin diffusion. We assume that $\alpha\left(X\right)$ and $\sigma\left(X\right)$, the drift and the diffusion coefficients are Lipschitz continuous. The solution X_t to the Langevin SDE is an Ito diffusion process and is Markov. Such an SDE is associated with the backward and forward Kolmogorov operators,

respectively denoted by L_t and L_t^*. L_t^* is known as the adjoint of L_t (see Equation A4.41, Appendix 4). Both these partial differential operators help in evaluating the system statistics and thus in obtaining weak solutions. Specifically, the adjoint L_t^* gives the PDE for the (transition) *pdf*, say, $\mathbb{f}_X(x|x_0)$. The PDE is known as the Fokker-Planck equation. For the Langevin SDE (4.103), the Fokker-Planck equation is given by:

$$\frac{\partial \mathbb{f}_X(x|x_0)}{\partial t} + \frac{\partial\left(a(X)\mathbb{f}_X(x|x_0)\right)}{\partial x} - \frac{1}{2}\frac{\partial^2\left(\sigma^2(x)\mathbb{f}_X(x|x_0)\right)}{\partial x^2} = 0 \tag{4.104}$$

The arguments x and x_0 in lower case is according to the convention generally followed in the definition of a *pdf*. The PDE (4.104) provides a means of obtaining the stationary or invariant *pdf*. That is, with $\frac{\partial \mathbb{f}_X(x|x_0)}{\partial t} \to 0$ as $t \to \infty$, one gets the stationary *pdf* denoted by, say, $\mathbb{f}_X(x)$ as the solution of the PDE:

$$\frac{d\left(a(x)\mathbb{f}_X(x)\right)}{dx} - \frac{1}{2}\frac{d^2\left(\sigma^2(X)\mathbb{f}_X(x)\right)}{dx^2} = 0 \tag{4.105}$$

with the boundary conditions:

$$\lim_{|x|\to\infty} \mathbb{f}_X(x) = 0, \int_{\mathbb{R}} \mathbb{f}_X(x)dx = 1 \tag{4.106}$$

Note that Ito diffusion processes described by the SDEs (as the one in Equation 4.103) may indeed converge to stationary *pdfs*. See Exercise 4.15 (in Exercises of Chapter 4) for an illustration on this aspect. We refer to Spencer and Bergmann (1993), Risken (1996) and von Wagner and Wedig (2000) for details on efficient methods to solve the ODE (4.105) *pdf* $\mathbb{f}_X(x)$. In the present context, we observe that for the case of additive noise with $\sigma \in \mathbb{R}$, the ODE (4.105) is satisfied if:

$$a(x) = -\frac{\sigma^2}{2}\nabla \log \mathbb{f}_X(x) \tag{4.107}$$

This observation shows that for the stationary *pdf* $\mathbb{f}_X(x)$ to match with the given target *pdf*, one can construct the Langevin SDE (4.103) with appropriate drift and diffusion coefficients. For instance, if $\sigma = \sqrt{2}$, the SDE is:

$$dX_t = -\nabla \log \mathbb{f}_X(X_t)\, dt + \sqrt{2} dB_t \tag{4.108}$$

Thus, in sampling problems, the overdamped Langevin SDE (4.108) is solved to obtain samples of $X \sim \mathbb{f}_X(x)$ that converges to the specified target *pdf*. An illustrative example of a sampling problem to draw samples of $X \approx \mathbb{f}_X(x)$ is given in Appendix 4 (item A.4). Suppose that the SDE is solved by the Euler-Maruyama (EM) method (item A4.3of Appendix 4) using a (uniform) time step Δt and initial condition X_0. $X(t)$ evolves according to a discrete EM sequence given by:

$$X_{k+1} = X_k - \nabla \log \mathbb{f}_X(X_k)\, \Delta t + \sqrt{2}\, \Delta B_k, \Delta B_k = B_k - B_{k-1} \tag{4.109}$$

The last equation throws light on the analogy between a sampling technique and optimization method. That is, if we have $f(x)$, a convex function having a continuous gradient, then with $f(x) = -\log f_X(x)$, the last equation corresponds to a line search in an optimization algorithm in a classical sense (akin to the algorithms discussed in Chapters 2 and 3). One significant difference is the presence of the extra noise term involving Brownian motion rendering the search intrinsically stochastic. This term facilitates exploration in the design space, thus avoiding a solution trapped near a local minimum. Hence, an analogous discrete approximation corresponding to an optimization problem is:

$$X_{k+1} = X_k - \nabla f\left(X_k\right)\Delta t + \sqrt{2}\,\Delta B_t \tag{4.110}$$

The algorithm pertaining to the last equation is referred to as the unadjusted Langevin algorithm (ULA) or Langevin Monte Carlo algorithm, where a Metropolis test (Appendix 3) may also be included at each iteration to accept or reject an update. With this modification, the method is known as Metropolis Adjusted Langevin Algorithm (MALA) (Welling and Teh 2011, Dubey *et al*. 2016). One may indeed consider it as the classical MALA. As with most stochastic algorithms based on heuristics like genetic algorithms (Section 3.4, Chapter 3), we start with a population of $N_p > 1$ particles or initial conditions. As the numerically integrated solution for a specific initial condition evolves according to Equation (4.110), one obtains a Markov chain. For large t, the transition *pdf* exhibits convergence to the stationary *pdf* $e^{-f(X)}$ and the first-order moment or ensemble sample mean $m_X = \dfrac{1}{N_p}\sum_{k=1}^{N_p} X_k$ approaches the optimum x^*. Thus, MALA in general is a class of MCMC methods in which the Markov Chain evolves as per the overdamped Langevin dynamics. The MALA, using the Langevin dynamics, proposes new moves, which are then accepted or rejected following the MH scheme. Since the Langevin dynamics involves gradient information of the target distribution, the method is more likely to move towards regions of high probability which is a major advantage over the use of largely arbitrary proposal distributions. This specific feature forms the basis for many of the applications of MCMC methods including optimization.

Both ULA and MALA show exponential convergence as studied in Roberts and Tweedie (1996a, 1996b). Readers are also referred to Hwang (1980), Gelfand and Mitter (1991) and Zhang et al. (2017) for more information on the convergence aspects. The convergence guarantee is with respect to the relative entropy or the KL divergence (Cheng and Bartlett 2018 and Vempala and Wibisono 2019). In the context of sampling, KL divergence is a measure of the 'distance' between the transition *pdf* corresponding to X evolving according to Equation (4.109) and the target *pdf*.

4.6 GEOMETRIC METHOD OF OPTIMIZATION BY RIEMANNIAN LANGEVIN DYNAMICS

Having identified that Equation (4.110) is a makeover of the familiar gradient descent step of classical optimization of an objective function $f(x)$, it is tempting to construct

a Riemannian version of gradient search (as in the geometric SDM), by just modifying the drift term in the discrete map (4.110) as:

$$X_{k+1} = X_k - \text{grad}f\left(X_k\right)\Delta t + \sqrt{2}\,\Delta B_t \tag{4.111}$$

grad $f(x)$ is the Riemannian gradient given in Equation (4.78) and defined on a tangent space $T_X M$. However, note that the Brownian motion B_t in the last equation is not appropriately restricted to be on M and hence the update X_{k+1} will also not remain on M. However, as a fix to this dilemma, the update at each iteration may be transferred to the manifold surface by exponential mapping as the solution evolves. In this way, Equation (4.111) may perhaps be considered to represent a Riemannian version of MALA (call it RMALA); see Li and Erdogdu (2020) who have used such a scheme for non-convex optimization as well as sampling. But the dilemma of inconsistency will still persist as $X_{k+1} - X_k$ is generally not a vector on $T_X M$. Indeed, as we shall show in the next chapter, both the drift and diffusion terms in the last SDE need a relook if a Riemannian update is sought. In any case, in the following example, we use RMALA (as it appears in Equation 4.111) to arrive at an optimal solution to the Ackley function, a benchmark function for optimization problems (Tang et al. 2009). Despite the theoretical inconsistency of RMALA, these results would serve to highlight the contrast in accuracy vis-à-vis more appropriate schemes to be considered in Chapter 5.

Example 4.9. The Ackley function is given by:

$$f\left(\boldsymbol{X}\right) = -a\exp\left(-b\sqrt{\frac{1}{n}\sum_{i=1}^{n}X_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(cX_i\right)\right) + a + \exp(1) \tag{4.112}$$

n is the dimension of the design variable $\boldsymbol{X}$ and $a, b, c \in \mathbb{R}$.

Solution. The n-dimensional surface corresponding to $f\left(\boldsymbol{X}\right): \mathbb{R}^n \to \mathbb{R}$ acts as the manifold M embedded in $\mathbb{R}^{n+1}$. The expression for grad$f\left(\boldsymbol{X}\right)$ is given in Appendix 4 (item A4.5) along with the Riemannian metric and Christoffel symbols. The metric is chosen to be similar to the one computed in Examples 4.5 and 4.6. The solution $\boldsymbol{X}_{k+1}$ in Equation (4.111) may be considered as an evolution over pseudo-time equivalent to a sequence of iterative steps. With a view to improving the exploration of the search space, we have adopted an annealing type approach (see Simulated Annealing (SA) method – Section 3.4.2, Chapter 3). To this end, an annealing type coefficient $\beta \in \mathbb{R}$ is introduced in the update strategy. The idea is to provide larger diffusion intensity to the update term in the initial stages of evolution and reduce it as the candidates approach the global optimum. The annealing-type coefficient β (with $1/\beta$ interpreted as the annealing temperature in SA) here appears as a scalar factor multiplying the update term so that the update equation becomes:

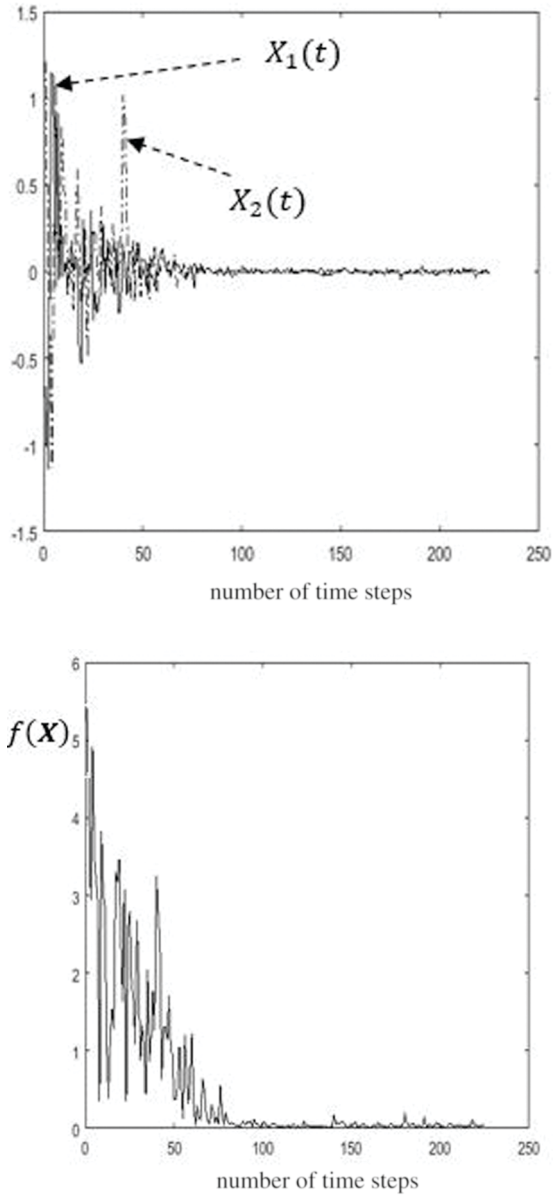
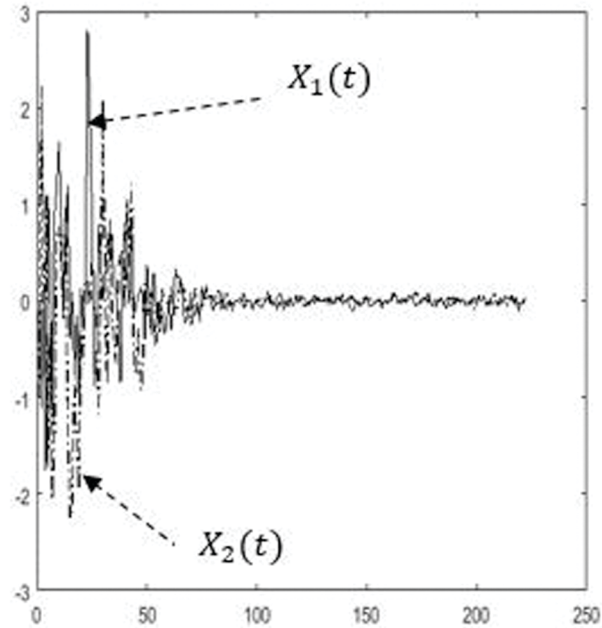


FIGURE 4.24a–b Optimization by RMALA of two-dimensional Ackley function: (a) evolution of the solution $X_1(t)$ (dark line) and $X_2(t)$ (dash-dot line) and (b) evolution of the objective function versus iterations, $\Delta t = 0.001$, $N_p = 10$. Optimization by RMALA of two-dimensional Ackley function: (c) evolution of the solution $X_1(t)$ (dark line) and $X_2(t)$(dash-dot line) and (d) evolution of the objective function versus iterations, $\Delta t = 0.01, N_p = 10$. Optimization by RMALA of two-dimensional Ackley function: (e) evolution of the solution $X_1(t)$ (dark line) and $X_2(t)$(dash-dot line) and (f) evolution of the objective function versus iterations, $\Delta t = 0.1$, $N_p = 10$.



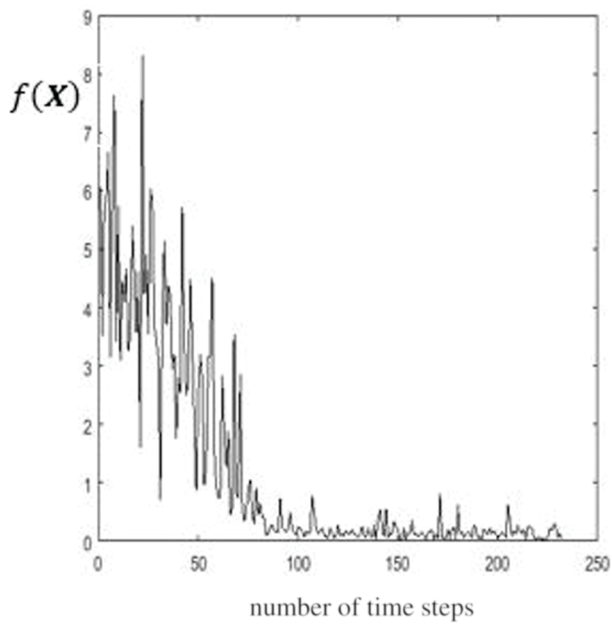
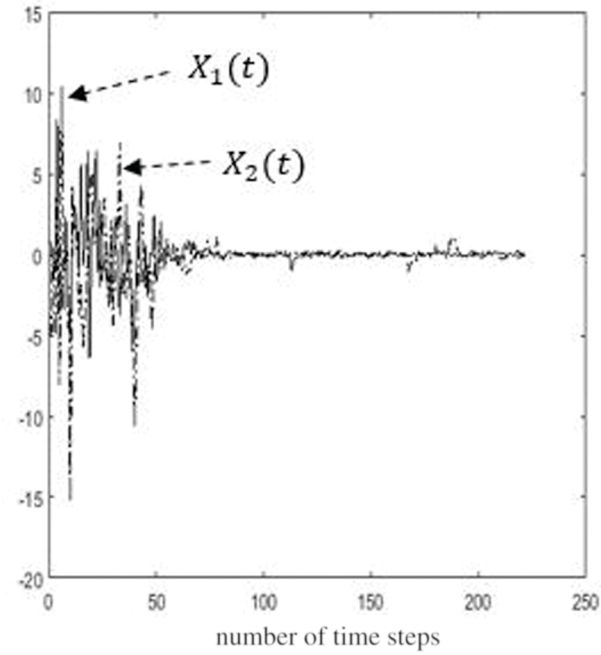


FIGURE 4.24c–d (Continued)



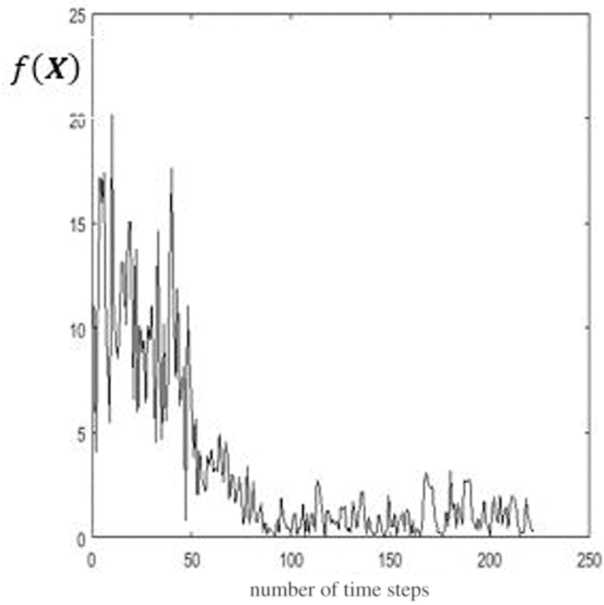


FIGURE 4.24e–f (Continued)

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k - \beta\left(\text{grad}f\left(\boldsymbol{X}_k\right)\right)\Delta t + \sqrt{2\beta}\,\Delta \boldsymbol{B}_t \tag{4.113}$$

Note that introduction of $\sqrt{\beta}$ in the noise term is consistent with the requirement in Equation (4.107). Similar to the control parameter 'time' in SA, β is reduced as the solution progresses.

Result by RMALA for the two-dimensional ($n = 2$) Ackley function is shown in Figures 4.24a–b. A time step of $\Delta t = 0.01$ is used in solving Equation (4.113). $N_p = 10$ is selected for the number of particles or the initial candidates. As the solution evolves, Metropolis test is performed at each iteration to accept or reject $\boldsymbol{X}_{k+1}$. The acceptable candidate solution corresponding to the minimum function value is stored at each time step. At the k^{th} step, if the update $\boldsymbol{X}_{k+1}$ is found successful after the MH acceptance test, it is transferred to the manifold. Exponential mapping (Section 4.2.7) is utilized for the purpose. The parameter β is initially chosen as 500 and reduced with time as $\beta_{k+1} = \beta_k \exp(0.01k)$ until β becomes less than 0.5. Figures 4.24a–f show results by RMALA with different time steps, going as high as a $\Delta t = 0.1$. While convergence is visibly poor for large Δt, all trials interestingly pass the Metropolis step.

Result by classical MALA is also obtained for the test function. The update is governed by the following equation:

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k - \beta \frac{df\left(\boldsymbol{X}\right)}{d\boldsymbol{x}}\bigg|_{\boldsymbol{X}_k} \Delta t + \sqrt{2\beta}\,\Delta \boldsymbol{B}_t \tag{4.114}$$

where $\dfrac{df(\boldsymbol{X})}{d\boldsymbol{x}}$ is the Euclidean first-order derivative. Parameters N_p and β are kept the same as in the geometric case. No convergence is observed for large $\Delta t = 0.1$. Convergence is found to be inconsistent for $\Delta t = 0.01$. The inconsistency is in the form of few successful runs out of repeated trials. Run with $\Delta t = 0.001$ is fully consistent (all trials pass the Metropolis step) and is shown in Figure 4.25.

■

CONCLUDING REMARKS

As a prelude to our exposition of geometric methods of optimization, we have given a gentle introduction to the theory of manifolds which forms the basis for these methods. While there exist numerous texts/monographs on the theory of manifolds, we have made the presentation limited to its application to optimization. Though the presentation is far from comprehensive, we have attempted to present the essential details in a style accessible to readers with an elementary knowledge in linear algebra and differential calculus.

Manifolds serve to generalize the notion of curves and surfaces in higher dimensions and are characterized by a locally Euclidean property. One needs differential geometry to capture the intrinsic structural properties of a manifold. Starting with the fundamental definition of a differentiable (smooth) manifold, we have presented the concept of a metric followed by the definitions of a few basic features: tangent space, tangent bundle and vector fields. The tangent space is the set of all tangent vectors at a point on the manifold M and is Euclidean. The tangent space is thus a vector space and the local coordinates x^i define a basis $\frac{\partial}{\partial x^i}$.

A manifold endowed with a metric – the invariant first quadratic form – is a basic construct in the description of a Riemannian manifold. For smooth real-valued functions on a Riemannian manifold, one can define the classical differential operators – the gradient and Hessian which are indeed useful in extending the classical methods of optimization to Riemannian versions (geometric methods). The extension in fact requires the additional notions of geodesics, connection, exponential mapping, and parallel transport on manifolds. These are described along with a few illustrative examples on optimization. The procedural steps are similar to classical methods except that at each iteration we move along the gradient on the tangent space at the current point and transfer the update to the manifold – possibly via the exponential map or a retraction-based approximation. In this context, the notion of 'connection' is significant. It helps us to define directional derivatives of vector fields – called the covariant derivatives – along curves and connects the tangent spaces at different locations of a manifold. The importance of the concept is more manifest when one implements conjugate gradient or quasi-Newton methods on manifolds where an update at the k^{th} iteration needs gradients belonging to different tangent spaces $T_{x_{k-1}} M$ and $T_{x_k} M$.

Following the importance of optimization strategies in statistical estimation problems (Section 3.2.2, Chapter 3), we have described how one may pose it as an optimization problem in a Riemannian setting. Here, we identify the manifold M as the statistical model comprising of the *pdf*s parameterized by the unknown variables in the given distribution. It is a Riemannian manifold with a metric given by the KL divergence which is shown to be equivalent to the FIM – the inner product on the manifold.

Highlighting the analogy between statistical sampling and optimization strategies, we describe an elegant geometric optimization scheme based on stochastic search using Langevin SDE. To understand the scheme based on the analogy, one needs a fair knowledge of SDEs and their solution methods. With a brief background provided in Appendix 4 on stochastic processes, associated SDEs and their numerical solution procedures, we have made use of the Langevin-based algorithm for function optimization problems. In a manifold setting, it amounts to solving the Langevin SDE on the tangent space at x_k to get an update using the Riemannian gradient and transferring the update to the manifold surface through exponential mapping.

We carry forward to the next chapter our discussion on geometric optimization schemes based on Langevin dynamics and focus on their possible improvements.

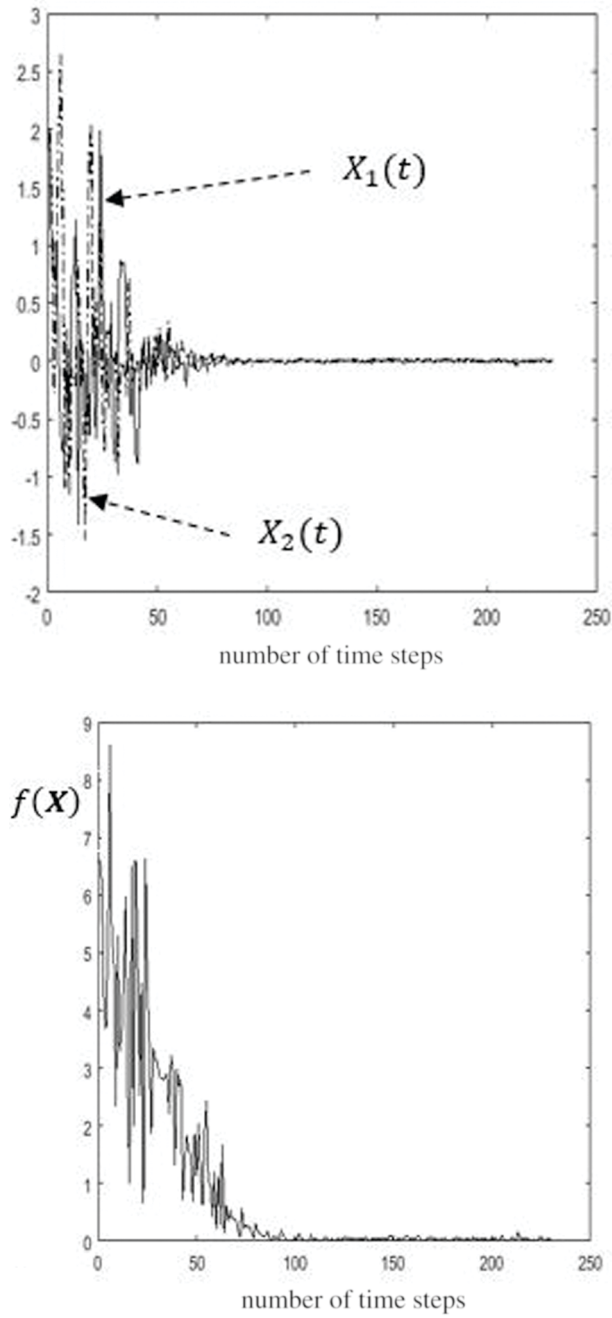


FIGURE 4.25 Optimization by classical MALA of two-dimensional Ackley function: (a) evolution of the solution $X_1(t)$(dark line) and $X_2(t)$(dash-dot line) and (b) evolution of the objective function versus iterations, $\Delta t = 0.001$, $N_p = 10$.

The improvement is in terms of stochastic development of an SDE on a manifold and tracing the evolution of design variables via the solution of the developed SDE. By this, we not only avoid the conceptual pitfalls of not working with the correct vectors on $T_x M$, but also avoid the computationally expensive step of exponential mapping at each iteration during the optimization process. An understanding of these improved schemes needs some appreciation of stochastic development on a manifold and, in particular, analysis of Brownian motion on a manifold.

EXERCISES

1. Show that the Riemannian metric g is invariant under coordinate transformation. [Hint: consider two parametrizations $\left(u^1, u^2\right)$ and $\left(\xi^1, \xi^2\right)$ with respective first quadratic forms $du^i \mathfrak{g}_{ij} du^j$ and $d\xi^i \tilde{\mathfrak{g}}_{ij} d\xi^j$. Show that $g = du^i \mathfrak{g}_{ij} du^j = d\xi^i \tilde{\mathfrak{g}}_{ij} d\xi^j$.]
2. Consider the two-dimensional space describing a cylinder of radius r by a coordinate chart. Find the metric g and the Christoffel symbols associated with the cylinder. The local coordinates are $x^1 = \theta$ and $x^2 = z$ and the surface coordinates of the cylinder are denoted by the transformation:

$$x = r\cos x^1 = r\cos\theta, y = r\cos x^1 = r\sin\theta \text{ and } z = x^2 = z.$$

3. Consider a manifold M with boundary ∂M. Show that the divergence operator satisfies the following product rule for a smooth function $f \in C^\infty\left(M\right)$:

$$\operatorname{div}\left(fX\right) = f \operatorname{div} X + \operatorname{grad} f, X \tag{E4.1}$$

and deduce the following 'integration by parts' formula:

$$\int_M \operatorname{grad} f, X \, dV == -\int_M f \operatorname{div} X \, dV + \int_{\partial M} f\left(X, N\right) d\tilde{V} \tag{E4.2}$$

where X is a vector field, N the outward unit normal to ∂M. dV and $d\tilde{V}$ are respectively the volume elements on M and of the induced metric on ∂M. See Lee (1997) for details on the divergence operator.

4. Consider a unit sphere S^2 with a curve $\gamma\left(t\right)$ being a circle at fixed latitude ϕ (see Figure 4.10). Find the total length of the curve. [Hint: use Equation (4.23) along with $g = du^i \mathfrak{g}_{ij} du^j$ in terms of the local coordinates.]
5. Find the surface area of the unit sphere S^2 in Exercise 4 above.
6. Consider two differentiable vector fields $X = X^i \dfrac{\partial}{\partial x^i}$ and $Y = Y^j \dfrac{\partial}{\partial x^j}$ on M. Show that the torsion $\tau\left(X, Y\right) = X^i Y^j \left(\Gamma^k_{ji} - \Gamma^k_{ij}\right) \dfrac{\partial}{\partial x^k}$ and hence note that the connection is symmetric if and only if $\Gamma^k_{ji} = \Gamma^k_{ij}$.
7. Prove that curvature on Riemannian manifold with Levi-Civita connection ∇ is:

$$R(X,Y)Z = \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z \tag{E4.3}$$

where $X = \dfrac{\partial}{\partial x^i}$ and $Y = \dfrac{\partial}{\partial x^j}$ are the local coordinate vector fields.

8. Refer to Equation (4.66) for the Riemannian curvature R and prove the following identities:

$$R(X, fY)Z = fR(X,Y)Z \text{ and}$$

$$R(fX,Y)Z = fR(X,Y)Z \tag{E4.4}$$

9. If $X = X^i \dfrac{\partial}{\partial x^i}$, $Y = Y^j \dfrac{\partial}{\partial x^j}$ and $Z = Z^k \dfrac{\partial}{\partial x^k}$, show that:

$$R(X,Y)Z + R(Z,X)Y + R(Y,Z)X = \nabla_X \left(\tau(Y,Z)\right) + \nabla_Y \left(\tau(Z.X)\right) + \nabla_Z \left(\tau(X,Y)\right)$$

$$+\tau\left(X,[Y,Z]\right) + \tau\left(Y,[Z,X]\right) + \tau\left(Z,[X,Y]\right) \tag{E4.5}$$

10. Solve the statistical estimation problem (Section 4.4) for a Gaussian *pdf* by RMALA. The unknown parameters are the mean and standard deviation denoted respectively by $\propto$ and σ. Assume availability of sufficient number (N) of observations. The Gaussian *pdf* is:

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \tag{E4.6}$$

11. Consider estimation of parameter σ of a Rayleigh *pdf* $f_X(x) =$. See Example 4.8 on MLE by the geometric method. Assume availability of (N) observations of the random variable X. Take the true parameter $\sigma = 2$.
12. Solve by RMALA (geometric version of MALA) for optimum of the two-dimensional Rosenbrock function $f(x) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2$ (also see Examples 4.6 and 4.7).

NOTATIONS

$\mathcal{A}$	collection of charts on a manifold
$\boldsymbol{B}_t \equiv \boldsymbol{B}(t)$	Brownian motion (noise)
$d(\boldsymbol{x}, \boldsymbol{y})$	distance metric
$\boldsymbol{d}_k$	direction vector

$D_v(f)$	directional derivative of a smooth function f
$\mathcal{D}(P(x) \| Q(x))$	KL divergence between two probability measures $P(x)$ and $Q(x)$
$e^i, i = 1,2,\ldots$	coordinate bases for T_p^*M
$E_i, i = 1,2,..$	coordinate bases for T_pM
$\mathbb{f}_Z(z;\boldsymbol{\theta})$	*pdf* of the observed data (Section 4.4)
$\mathbb{f}_X(x\|x_0)$	transition *pdf* (Equation 4.104)
$\mathbb{f}_X(x)$	target *pdf* (stationary *pdf* of Langevin equation)
$F(x,y)$	an injective scalar function
$\boldsymbol{F}(\boldsymbol{x},\boldsymbol{y})$	an injective vector function
$\mathbb{F}_Z(z;\theta)$	probability distribution (CDF) of the observed data (Section 4.4)
g	Riemannian metric
$\mathfrak{g}$	symmetric matrix associated with g
$g_p : T_pM \times T_pM \to \mathbb{R}$	symmetric bilinear map (inner product)
grad	Riemannian gradient
$\mathcal{G}$	a differentiable map
Hess$f(\boldsymbol{x})$	Riemannian Hessian of $f(\boldsymbol{x})$ at $\boldsymbol{x}$ on a manifold
I	arc length of the curve over the interval $[a,b]$ (Equation 4.23)
$\boldsymbol{I}_n(\boldsymbol{\theta})$	Fisher information matrix (FIM)
J	Jacobian matrix
$l(\boldsymbol{\theta};z)$	log-likelihood function (Section 4.4)
$L(\boldsymbol{\theta};z)$	likelihood function (Section 4.4)
N_p	number of particles or the initial candidates
$R(.,.)$	a differential operator associated with the Riemannian curvature (Equation 4.66)
S	a (curved) surface
S^2	two-dimensional sphere manifold
TM	tangent bundle on a manifold – the set $\{T_pM \mid p \in M\}$
$T_p(M)$	tangent space at the point p on the manifold M
$T_p(\mathbb{R}^n)$	tangent space at the point p in $\mathbb{R}^n$
T_p^*M	cotangent space on the manifold M
Γ_{ij}^k	Christoffel symbols (Equation 4.39)
U	an open neighbourhood on a manifold
v	vector in $T_p(M)$
$v(t)$	velocity of a particle (Equation 4.2)
V	vector space in a Euclidean space

$w \in T_p^* M$	covector
$w(t)$	randomly fluctuating force – white noise (Equation 4.3)
$\{x_1(t), x_2(t), \ldots, x_n(t)\}$	coordinate functions at any point on a curve $\gamma(t)$
$X(p) \in T_p M$	tangent vector on a manifold M
$\boldsymbol{X}_t \equiv \boldsymbol{X}(t)$	a stochastic process
$\boldsymbol{z}$	observation data (Section 4.4)
$\boldsymbol{Z}$	vector of random variables (Section 4.4)
$\pm$	parameter in the generalized exponential *pdf* (Equation 4.98)
$\alpha(X)$	drift term in the SDE (4.103)
β	annealing-type coefficient
β_k	parameter in CG method (Equation 4.85)
Δt	time step
∇	linear connection on M
$\nabla f(x)$	gradient of the function $f(x)$
∇f_p	gradient of the function f at p on a manifold
φ	a bijective map between a manifold and a Euclidean space
Ψ	a bijective map like φ
ϕ	polar angle (Figure 4.10)
θ	azimuth angle (Figure 4.10)
$\boldsymbol{\theta} = (\theta^1, \theta^2, \ldots)$	parameter set in the *pdf* $\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z}; \boldsymbol{\theta})$
λ	parameter in the generalized exponential *pdf* (Equation 4.98)
$\sigma(X)$	diffusion term in the SDE (4.103)
$\gamma(t)$	a smooth curve on a manifold
$\chi : C^\infty(\mathbb{R}^n) \to \mathbb{R}$	linear map satisfying the product rule (the Lebnitz property)
$Exp_p(v)$	exponential map
$Exp_p^{-1}(q)$	logarithmic map
$[X, Y]$	Lie bracket (Equation 4.56)

REFERENCES

Abbas Abdolmaleki, A., Rudolf Lioutikov, R., Nuno Lau, N., Reis, L. P., Peters J., and Neumann, G. 2015. Model-based relative entropy stochastic search. Advances in Neural Information Processing Systems 28 (NIPS 2015).

Absil, P.A., C. G. Baker, and K. A. Gallivan. 2004. Trust-region methods on Riemannian manifolds with applications in numerical linear algebra. *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems (MTNS2004)*. Leuven, Belgium.

Absil, P. A., C. G. Baker, and K. A. Gallivan. 2007a. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics* 7(3): 303–330.

Absil, P., A. R. Mahony, and R. Sepulchre. 2007b. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press.

Alspach, D. and Sorenson. H. 1972. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control* 17(4): 439–448.

Amari, S. 1997. Neural learning in structured parameter spaces — natural Riemannian gradient. In: Advances in Neural Information Processing Systems *9*. The MIT Press, Cambridge, MA, pp. 127–133.

Amari, S. 1983. Differential geometry of statistical inference. In: *Probability Theory and Mathematical Statistics*, K. Ito and J. V. Prokhorov (eds.), pp. 26–40. Springer Lecture Notes in Math.

Amari, S.I. 1985. *Differential Geometrical Methods in Statistics*. Springer Lecture Notes in Statistics, 28, Springer.

Aubram, D. 2009. *Differential Geometry Applied to Continuum Mechanics*. Shaker Verlag. Germany.

Azizi, A. and Yazdi, P. G. 2019. *White Noise: Applications and Mathematical Modeling*. Springer Nature Singapore Pte Ltd.

Baker, C. G. 2008. *Riemannian Manifold Trust-Region Methods with Applications to Eigenproblems*. PhD thesis. Florida State University. Tallahassee, Florida.

Baker, C. and Parks, G. T. 2016. Riemannian optimization and multidisciplinary design optimization. *Optimization and Engineering* 17: 663–693.

Bento, G. C. and Melo, J. G. 2012. Subgradient method for convex feasibility on Riemannian manifolds. *Journal of Optimization Theory and Applications* 152: 773–785

Boothby, 1975. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, NY.

Botsaris, C.A. 1981a. A class of differential descent methods for constrained optimization. *Journal of Mathematical Analysis and Applications* 79: 96–112.

Botsaris, C.A. 1981b. Constrained optimization along geodesics. *Journal of Mathematical Analysis and Applications* 79: 295–306.

Boumal. N. 2014. *Optimization and Estimation on Manifolds*. PhD thesis, Université catholique de Louvain.

Boumal, N., Mishra, B., P.-A. Absil, P. A. and Sepulchre, R. 2014. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research* 15(42): 1455−1459.

Boyat, A. K. and Joshi, B. K. 2015. A review paper: noise models in digital image processing. *Signal & Image Processing: An International Journal (SIPIJ)* 6(2): 63–75.

Brown, R. 1827. *A Brief Account of Microscopical Observations*. London (not published).

Cayton. L. 2005. *Algorithms for Manifold Learning*. University of California at San Diego Tech. Rep 12, pp. 1–17.

Cheng, X. and Bartlett, P. 2017. *Convergence of Langevin MCMC in KL-divergence*. In: Proceedings of the 29th International Conference on Algorithmic Learning Theory (ALT), pp. 186–211.

Christian, A. 2015. *Differential Geometry: Curvature and Holonomy*. PhD thesis. The University of Texas at Tyler.

Christoffel, E. B. 1869. *Journal für die reine und angewandte Mathematik* 70: 46–70.

Clough, R. W. and Penzien. J. 1982. *Dynamics of Structures*. McGraw-Hill.

Cohen, L. 2005. The history of noise. *IEEE Signal Processing Magazine* 22(6): 20–45.

Cowan, G. 1998. *Statistical Data Analysis*. Oxford University Press. NY.

da Cruz Neto, J. X, Ferreira, O. P, Lucambio P´erez, L. R and N´emeth, S. Z. 2006. Convex- and monotone-transformable mathematical programming problems and a proximal-like point method. *Journal of Global Optimization* 35: 53–69.

Dalalyan, A. S. 2017a. Further and stronger analogy between sampling and optimization: Langevin Monte Carlo and gradient descent. Conference on Learning Theory. *Proceedings of Machine Learning Research* 65: 678–689.

Dalalyan, A. S. 2017b. Theoretical guarantees for approximate sampling from a smooth and log-concave density. *Journal of the Royal Statistical Society B* 79(3): 651–676.

Do Carmo. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall.

Dreisigmeyer, D. W. 2007. *Equality Constraints, Riemannian Manifolds and Direct Search Methods*. Technical report. Los Alamos National Laboratory Los Alamos, NM.

Du, K. and Swamy, M. N. S. 2014. *Neural Networks and Statistical Learning*. 1st Ed. Springer-Verlag. London.

Dubey, K. A., Reddi, S. J., Williamson, S. A., Poczos, B., Smola, A. J. and Xing. E. P. 2016. Variance reduction in stochastic gradient Langevin dynamics. In: *Advances in Neural Information Processing Systems*, pp. 1154–1162.

Durmus, A., Majewski, S. and Miasojedow, B. 2019. Analysis of Langevin Monte Carlo via convex optimization. *Journal of Machine Learning Research* 20: 1–46.

Edelen, D.G.B. 1985. Applied Exterior Calculus. John Wiley & Sons, Inc., NY.

Einstein, A. 1905. On the movement of small particles suspended in stationary liquids required by the molecular-kinetic theory of heat. *Annals of Physics* 17: 549–60. [Appears in *The Collected Papers of Albert Einstein*. English translation by Anna Beck ~Princeton U.P., Princeton, NJ, 1989. 2. 123–134.]

Fletcher, R. and C. M. Reeves. 1964. Function minimization by conjugate gradients. *Computer Journal* 7 : 149–154.

Fong R. S. and Tino. P. 2019. Extended stochastic derivative-free optimization on Riemannian manifolds. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 257–258. ACM.

Gabay, D. 1982. Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications* 37(2): 177–219.

Gauss, K. F. 1827. General investigations of curved surfaces of 1827 Presented to the Royal Society, October 8. Translated with notes and a bibliography by Morehead, J. C., and A. M. Hiltebeitel. *Fellows in Mathematics in Princeton University*. Copyright, 1902, by The Princeton University Library.

Gelfand, S. B. and Mitter, S. K. 1991. Recursive stochastic algorithms for global optimization in R^d . *SIAM Journal on Control and Optimization* 29(5): 999–1018.

Gray, R. M. 1990. *Entropy and Information Theory*. Springer Verlag, NY.

Gray, R. M. and Davisson, L. D. 2004. *An Introduction to Statistical Signal Processing*. Cambridge University Press.

Hajri, H., Said, S. and Berthoumieu, Y. 2017. Maximum likelihood estimators on manifolds. HAL-open science 01500284.

Hauberg, S., Lauze, F. and Pedersen, K. S. 2013. Unscented Kalman filtering on Riemannian manifolds. *Journal of Mathematical Imaging and Vision* 46(1): 103–120.

Hegde, C. 2012. *Nonlinear Signal Models: Geometry, Algorithms, and Analysis*. PhD thesis. Rice University.

Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Ann Arbor. MI.

Hosseini, R. and Sra, S. 2015. Matrix manifold optimization for Gaussian mixtures. In: Advances in Neural Information Processing Systems (NIPS) 28: 910–918.

Hsu, E. P. 2002. *Stochastic Analysis on Manifolds*. *Graduate studies in Mathematics.* American Mathematical Society.

Hwang, C. R. 1980. Laplace's method revisited: weak convergence of probability measures. *The Annals of Probability* 1177–1182.

Huang, W. 2013. *Optimization Algorithms on Riemannian Manifolds with Applications*. PhD thesis. Florida State University.

Ito, K. .1951. On stochastic differential equations. *Memoirs of the American Mathematical Society* 4: 1–51.

Jazwinski, A. H. 1970. *Stochastic Processes and Filtering Theory*. Academic Press, NY.

Jhonson, D. H. 2013. *Statistical Signal Processing*. Lecture notes. Rice University, Houston, TX.

Karatzas, I. and Shreve. S. 1991. *Brownian Motion and Stochastic Calculus*. Springer-Verlag. NY.

Klebaner, F.C. 1998. *Introduction to Stochastic Calculus with Applications*. Imperial College Press, London.

Kloeden, P. E. and Platen, E. 1992. *Numerical Solution of Stochastic Differential Equations.* Springer-Verlag. Berlin.

Kubo, R. 1986. Brownian motion and nonequilibrium statistical mechanics. *Science* 233(4761): 330–334.

Kullback, S. and Leibler, R. A. 1951. On information and sufficiency. *Annals of Mathematical Statistics* 22(1): 79–86.

Lang, S. 1995. *Differential and Riemannian Manifolds*. Graduate Texts in Mathematics. Springer-Verlag.

Langevin, P. 1908. Sur la théorie de mouvement brownien. *Comptes rendus de l'Académie des Sciences*146: 530. (English translation: Langevin, P. 1997. On the theory of Brownian motion. *American Journal of Physics* 65: 1079.)

Lee. J. M. 1997. *Riemannian Manifolds: An Introduction to Curvature*. Springer-Verlag, NY.

Lee. J. M. 2003. *Introduction to Smooth Manifolds*. 218 of Graduate Texts in Mathematics. Springer-Verlag, NY.

Levi-Civita, T. 1917. The notion of parallelism on any manifold. *Rendiconti Circolo Matematico del di Palermo* [in Italian] 42: 173–205.

Li, M. B. and Erdogdu, M. A. 2020. Riemannian Langevin algorithm for solving semidefinite programs. arXiv preprint arXiv:2010.11176

Lin T. and H. Zha. 2008. Riemannian manifold learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(5): 796–809.

Luenberger, D. G. 1972. The gradient projection method along geodesics. *Management Science* 18: 620–631.

Ma, Y. and Fu, Y. (eds.) 2011. *Manifold Learning Theory and Applications*. CRC Press. London.

Manton, J. H. 2013. A primer on stochastic differential geometry for signal processing. *IEEE Journal of Selected Topics in Signal Processing* 7(4): 681–699.

Mamajiwala, M. and Roy, D. 2022. Stochastic dynamical systems developed on Riemannian manifolds. *Probabilistic Engineering Mechanics* 67(5).

Martin, J., Wilcox, C. L., Burstedde, C. and Ghattas, O. 2012. A stochastic Newton MCMC method for large scale statistical inverse problems with application to seismic inversion. *SIAM Journal on Scientific Computing* 34(3): 1460–1487.

Maruyama, G. 1955. Continuous Markov processes and stochastic equations. *Rendiconti del Circolo Matematico di Palermo* 4: 48–90.

Milstein, G. N. 1974. Approximate integration of stochastic differential equations. *Theory of Probability & Its Applications* 19(3): 557–562.

Murray. M. K. 1996. Bundle Gerbes. *Journal of the London Mathematical Society* 54: 403–416.

Nigam, N. C. and Narayanan, S. 1994. *Applications of Random Vibrations*. Springer-Verlag.

Øksendal, B. 2003. *Stochastic Differential Equations: An Introduction with Applications*. Springer-Verlag, Berlin.

Ring, W. and Wirth, B. 2012. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM Journal on Optimization* 22(2): 596–627.

Risken, H. 1996. Fokker-Planck Equation. In: *The Fokker-Planck Equation*. Springer Series in Synergetics, vol 18. Springer, Berlin, Heidelberg

Roberts, G.O. and Tweedie, R.L. 1996a. Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika* 83: 96–110.

Roberts, G. O. and Tweedie, R. L. 1996b. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli* 2(4): 341–363.

Rogers, L. C. G. and Williams, D. 2000. *Diffusions, Markov Processes, and Martingales: Volume 1, Foundations*. 2nd Ed. Cambridge University Press. UK.

Rosen, J. B. 1960. The gradient projection method for nonlinear programming. Part I. Linear constraints. *SIAM Journal on Applied Mathematics* 181–217.

Rosen, J. B. 1961. The gradient projection method for nonlinear programming, Part 2: Nonlinear constraints. *SIAM Journal on Applied Mathematics* 9: 514–553.

Roy, D. and Rao, G. V. 2012. *Elements of Structural Dynamics: A New Perspective*. John Wiley & Sons. UK.

Roy, D. and Rao, G. V. 2017. *Stochastic Dynamics, Filtering and Optimization*. Cambridge University Press, UK.

Rudin, W. 1976. *Principles of Mathematical Analysis*. 3rd ed. NY: McGraw-Hill.

Sato, 2013. *Riemannian Optimization Algorithms and Their Applications to Numerical Linear Algebra*. PhD thesis. Kyoto University, Japan.

Simmons, G. F. and Krantz, S. G. 2006. *Differential Equations: Theory, Technique, and Practice*. McGraw-Hill.

Smith, S. T. 1993. *Geometric Optimization Methods for Adaptive Filtering*. PhD thesis. Harvard University. Cambridge, Massachusetts.

Smith, S. T. 1994. Optimization techniques on Riemannian manifolds. *Fields Institute Communications* 3.

Smith, S. T. 2005. Covariance, subspace, and intrinsic Cramer-Rao bounds. *IEEE Transactions on Signal Processing* 53(5): 1610–1630.

Smith, A. F. M. and G. O. Roberts, G. O. 1993. Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society. Series B (Methodological)* 3–23.

Spencer Jr. B. F. and L. A. Bergman. 1993. On the numerical solution of the Fokker-Planck equation for nonlinear stochastic systems. *Nonlinear Dynamics* 4(4): 357–372.

Stratonovich, R. L. 1966. A new representation for stochastic integrals and equations. *SIAM Journal on Control and Optimization* 4: 362–371.

Tang, K., X. Li, P. N. Suganthan, Z. Yang, and T. Weise. 2009. Benchmark functions for the CEC'2010 special session and competition on large scale global optimization. Nature Inspired Comput. Applicat. Lab., Tech. Rep.

Tu, L.W. 2011. *An Introduction to Manifolds*. 2nd Ed. Springer Science+ Business Media, LLC.

Turaga, P., A. Veeraraghavan, and R. Chellappa. 2008. Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8.

Vanmarcke, E. H. 1983. *Random Fields*. Cambridge. MA. MIT Press.

Vecer, J. 2011. *Stochastic Finance: A Numeraire Approach*. CRC Press. NY.

Vempala, S. and Wibisono, A. 2019. Rapid convergence of the unadjusted Langevin algorithm: Isoperimetry suffices. Advances in Neural Information Processing Systems 32: 8094–8106.

von Wagner, U. and W. V. Wedig. 2000. On the calculation of stationary solutions of multi-dimensional Fokker-Planck equations by orthogonal functions. *Nonlinear Dynamics* 21(3): 289–306.

Wiener, N. 1923. Differential space. *Journal of Mathematical Physics* 58: 131–174.

Welling, M. and Y. W. Teh. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In: *Proceedings of the International Conference on Machine Learning*, pp. 681–688.

Wibisono, A. 2018. Sampling as optimization in the space of measures: The Langevin dynamics as a composite optimization problem. *Proceedings of Machine Learning Research* 75: 1–35.

Wu, F. and Hu, S. 2008. Stochastic functional Kolmogorov-type population dynamics. *Journal of Mathematical Analysis and Applications* 347: 534–549.

Zhang, Y., Liang, P. and Charikar, M. 2017. A hitting time analysis of stochastic gradient Langevin dynamics. Proceedings of Machine Learning Research 65: 1–43.

Zwanzig, R. 2001. *Nonequilibrium Statistical Mechanics*. Oxford University Press, Inc. NY.

BIBLIOGRAPHY

Chandrasekhar, S. 1943. Stochastic problems in physics and astronomy. *Reviews of Modern Physics* 15(1): 1–89.

Christian, A. 2015. *Differential Geometry: Curvature and Holonomy*. Master of Science thesis. The University of Texas at Tyler. USA.

Cicotti, G. and J. P. Ryckaert, 1981. On the derivation of the generalized Langevin equation for interacting Brownian particles. *Journal of Statistical Physics* 26(1): 73–82.

Ferreira, O. P., M. S. Louzeiro, L. F. Prudente. 2018. Gradient method for optimization on Riemannian manifolds with lower bounded curvature. *SIAM Journal on Optimization*. Vol. 29(4). pp. 2423–3230.

Hanggi, P. and P. Talkner, 1978. On the equivalence of time-convolution less master equations and generalized Langevin equations. *Physics Letters A* 68(1): 9–11.

Neto, J. X. da C., L. L. de Lima, and P. R. Oliveira. 1998. Geodesic algorithms in Riemannian geometry. *Balkan Journal of Geometry and Its Applications* 3(2): 89–100.

5 Stochastic Analysis on a Manifold and More on Geometric Methods

5.1 INTRODUCTION

While discussing the geometric methods based on Langevin dynamics in the last chapter, a mention has been made about a possible improvement in the optimization schemes by developing the governing dynamics (e.g. in the form of an SDE) on the manifold. With this in view, we continue in this chapter to provide a geometrically consistent strategy to stochastically develop the Langevin SDE on the Riemannian manifold with a suitably constructed metric and the associated connection. We make use of concepts from differential geometry (Hsu 2002) and stochastic calculus on manifolds to geometrically adapt an SDE. This is usually known as stochastic development on manifolds and described in Section 5.2. We apply this specifically to the Langevin diffusion equation to obtain a Geometrically Adapted Langevin Algorithm (GALA being the acronym) which in turn is used as a tool for optimization. Solution by GALA for some test objective functions is given in Section 5.3 of this chapter. In this context, an intuitive understanding of the concept of stochastic development is possible if we think of the following exercise (Manton 2013) performed on a sphere $S^2: \left\{(x,y,z) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 = 1\right\}$. For instance, let us consider a simple two-dimensional SDE:

$$dX(t) = \left(dB_1(t),\ dB_2(t)\right)^T \tag{5.1}$$

The SDE corresponds to simultaneous evolutions of two statistically independent Brownian motions $B_i(t), i = 1,2$. Solving the SDE numerically by the EM method over an interval (0–0.5 s) and with a step size $\Delta t = 0.01$ s and plotting the two Brownian motions as a 2-D graph, we have Figure 5.1a. Now, the stochastic development of the SDE on S^2 may be construed as rolling the sphere (without slipping) along the 2-D graph and to get the path traced on the sphere. That is, with the north pole on the sphere kept at the initial point $(0,0)^T$ in the 2-D graph, the sphere is rolled along the first line segment till its end point. Assume that the path is imprinted on the sphere as it rolls. Without lifting the sphere, let the process be repeated for each line segment to get a path fully traced. The path so obtained on S^2 may be considered as a solution to the stochastically developed SDE on the manifold. Here, to mathematically encode this exercise, we generate a geodesic path on S^2 corresponding to the k^{th} line segment of the 2-D graph with a scaled velocity equal to $X_{k+1} - X_k$ where $X_k := X(t_k)$. In other words, geodesic path over an interval Δt corresponding to a

DOI: 10.1201/9781003414063-5

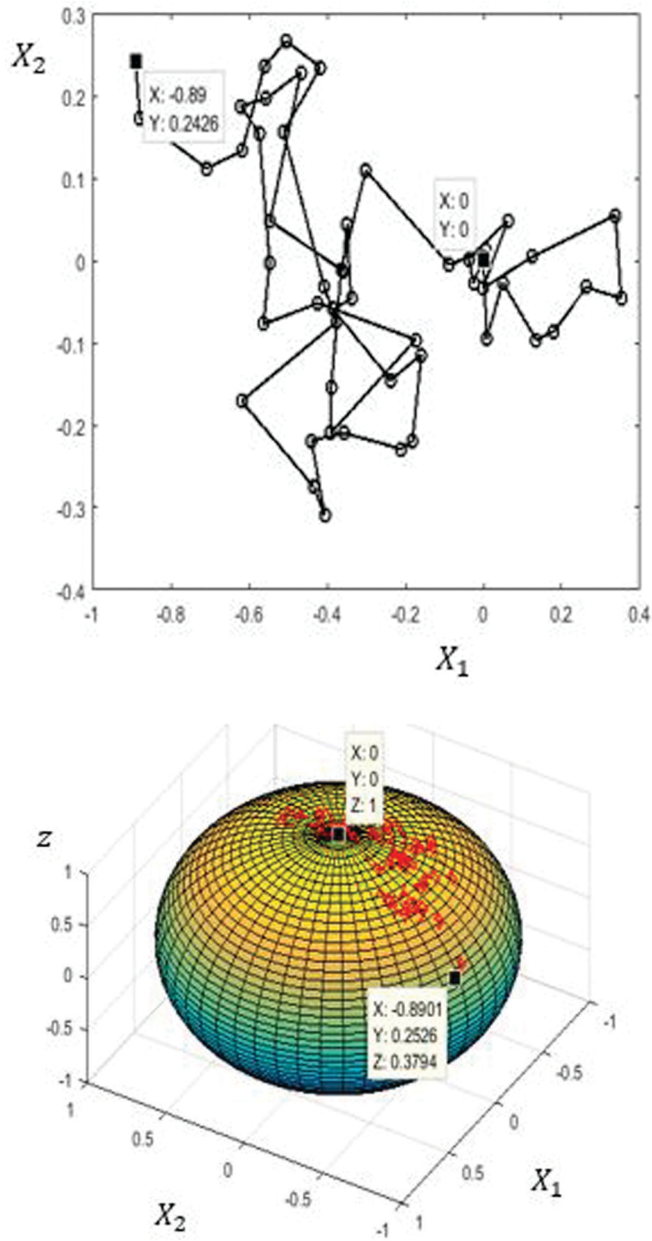


FIGURE 5.1 An exercise to simulate stochastic development on a manifold (here the sphere S^2 : (a) a trajectory of two-dimensional Brownian motion obtained numerically by solving the SDE (5.1) over the interval (0 – 0.5 s) with $\Delta t = 0.01$ (starting at point $(0,0)^T$ and ending at point $(-0.89, 0.2426)^T$ (marked by black squares in the figure); (b) simulated solution on a sphere S^2 starting from the north pole (0, 0, 1) and ending at point $(-0.8901, 0.2526, 0.3794)^T$ – (here indirectly obtained by solving the geodesic Equation 4.38).

line segment is obtained by solving Equation (4.38) (Chapter 4). The path so obtained on the manifold is shown in Figure 5.1b.

We will further discuss the concept of stochastic development in Section 5.2. This is followed in Section 5.3 by an application of this concept (as embedded in GALA) to optimization problems which is the primary objective in this chapter. We expect relatively fast and efficient results by GALA, specifically in solving large dimensional optimization problems compared to RMALA, a Riemannian version of MALA presented in Chapter 4 (Section 4.6).

Another significant application of GALA to MCMC simulation problems with emphasis on parameter estimation of probability distribution models is presented in Section 5.4. Readers may recall that statistical estimation is essentially an optimization problem (as illustrated in Chapters 3 and 4). The geometric SDM discussed in Chapter 4 and applied to the parameter estimation problem is similar to RMALA save for the absence of the noise term. Notice that the Riemannian gradient $\text{grad}\left(l\left(\boldsymbol{\theta}_k; \boldsymbol{Z}\right)\right)$ in Equation (4.99) of the log-likelihood function is the drift coefficient in the Langevin SDE (4.103). The parameter estimation example 4.8 is reconsidered in Section 5.4 for obtaining the result by GALA. In addition, application to a large dimensional problem is also presented to showcase its relative performance of GALA over RMALA. For notational simplicity, we do not indicate vectors and tensors by boldfaced symbols in this chapter. In addition, the reader will note that the coordinate components, say of a point x on an m-dimensional manifold M, are now denoted using superscripts, i.e. $x = (x^1, x^2, \ldots, x^m)$.

5.2 STOCHASTIC DEVELOPMENT ON A MANIFOLD

While presenting the geometric methods of optimization in Chapter 4, we assumed that Riemannian manifold was embedded within a higher dimensional Euclidean space, which is ensured by Whitney's embedding theorem (Cohen 1985). Characterizing the embedding space in general is however no trivial task. Instead, by exploiting the principle of stochastic development (Hsu 2002), we may exploit a fully intrinsic description of a Riemannian manifold, thus bypassing the problematic issue of embedding within an ambient Euclidean space. Also, in optimization problems, it facilitates a solution for an optimum on the manifold without the computationally intensive step of exponential mapping (step 4 of Table 4.1, Chapter 4).

As a precursor to the concept of stochastic development, we wish to make a brief mention of the Laplace-Beltrami (LB) operator (Hsu 2002) which is an analogue of the familiar Laplacian operator Δ_E in $\mathbb{R}^n$. The LB operator is a fundamental geometric object to a manifold in that it exhibits many desirable characteristics. It is a linear self-adjoint elliptic operator whose eigenfunctions may be used as a natural basis for square integrable functions on a manifold similar to the Fourier representation of periodic functions. It is extensively used for various tasks: surface reconstruction, shape representation and interpolation, mesh processing including spectral analysis on discrete surfaces (Rustamov 2007, Liu *et al.* 2008, Belkin and Niyogi 2001, Levy and Zhang 2010).

The Laplacian operator Δ_E is given by divergence of the gradient vector field ∇f of some function $f : C^2\left(\mathbb{R}^n\right)$. Thus, with Einstein summation convention implied, one has:

$$\Delta_E f = \operatorname{div}(\nabla f) = \frac{\partial^2 f}{\partial x^i \partial x^i} \tag{5.2}$$

As is evident, this is equivalent to the trace of the Hessian, i.e.:

$$\Delta_E = \operatorname{trace}\left[\frac{\partial^2 f}{\partial x^i \partial x^j}\right], i, j = 1, 2, \ldots n \tag{5.3}$$

Recall that the infinitesimal generator* L_t of Brownian motion is $\frac{1}{2}\frac{\partial^2 f}{\partial x^i \partial x^i}$ (Roy and Rao 2017). Hence, one may relate the Laplacian to the infinitesimal generator of the Euclidean Brownian motion as $L_t = \frac{\Delta_E}{2}$. An obvious extension is the relation that LB operator has with a Brownian motion on a manifold which provides a pointer towards the associated SDE on a manifold. Given a Riemannian manifold (M, g), the LB operator $\Delta_M : C^\infty(M) \to C^\infty(M)$ is a divergence operator analogous to Δ_E and is related to covariant derivatives on the manifold. Thus, for C^∞ real valued functions f:

$$\Delta_M f = \operatorname{div}_M(\operatorname{grad} f) \tag{5.4}$$

$\operatorname{grad} f$ is a vector field on M given by Equation (4.78) with $f \in C^\infty(M)$. Having defined in Equation (4.48) the covariant derivative in terms of local coordinates and with $E_i = \frac{\partial}{\partial x^i}, i = 1, 2, ..$ being the orthonormal bases for $T_p M$, the divergence of a vector field $X = X^i E_i \in T_p(M)$ at a point $p \in M$ is given by:

$$\operatorname{div}_M(X) = \nabla_{E_i}\left(X^i E_i\right) = \left(E_i X^i + X^i \Gamma^k_{ik}\right) \tag{5.5}$$

---

* infinitesimal generator

The backward Kolmogorov operator L_t in Equation (A4.41) of Appendix 4 is also known as infinitesimal generator which is rewritten hereunder:

$$L_t = \sum_{i=1}^{m} a_i(t, X_t)\frac{\partial}{\partial x_i} + \frac{1}{2}\sum_{i,j=1}^{m}\sum_{l=1}^{n}\sigma_{il}(t, X_t)\sigma_{jl}(t, X_t)\frac{\partial^2}{\partial x_i \partial x_j} \tag{i}$$

L_t is the generator of the vector diffusion process X_t which is the solution to the vector SDE:

$$dX_t = a(t, X_t)dt + \sigma(t, X_t)dB_t \tag{ii}$$

where Einstein summation convention is implied. Note that, in $\mathbb{R}^n$, the divergence $\Delta_E = E_i X^i$, the first term on the LHS of the last equation. Now, it follows that the LB operator Δ_M is:

$$\Delta_M(f) = \mathrm{div}_M(\mathrm{grad}\, f) = E_i\left((\mathrm{grad}\, f)^i\right) + \Gamma_{ik}^k (\mathrm{grad}\, f)^i \tag{5.6}$$

Following the formulae for Christoffel symbols, Γ_{ik}^k can be expressed as:

$$\Gamma_{ik}^k = \frac{\partial\left(\log\sqrt{|\mathfrak{g}|}\right)}{\partial x^k} = E_k\left(\left(\log\sqrt{|\mathfrak{g}|}\right)\right) \tag{5.7}$$

With $(\mathrm{grad}\, f)^i = \mathfrak{g}^{ij} E_j f$, Equation (5.6) becomes (Urakawa 1993, Hsu 2002):

$$\Delta_M(f) = E_i\left(\mathfrak{g}^{ij} E_j f\right) + E_k\left(\left(\log\sqrt{|\mathfrak{g}|}\right)\right)\mathfrak{g}^{ij} E_j f$$

$$= \frac{1}{\sqrt{|\mathfrak{g}|}} E_i\left(\sqrt{|\mathfrak{g}|}\ \mathfrak{g}^{ij} E_j f\right)$$

$$= \mathfrak{g}^{ij}\left(\frac{\partial^2 f}{\partial x^i \partial x^j} - \Gamma_{ij}^k \frac{\partial f}{\partial x^k}\right) \tag{5.8}$$

Now, taking cue from the relationship $L_t = \Delta_M(f)/2$ between the operator and the infinitesimal generator, one may show that the Brownian process evolving on a Riemannian manifold is the solution to the SDE:

$$d\hat{X}_t^k = -\frac{1}{2}\mathfrak{g}^{ij}\Gamma_{ij}^k dt + \sqrt{\mathfrak{g}^{kj}}\ dB_t^j,\ k = 1,2,\ldots,n \tag{5.9}$$

Solution of the SDE (5.9) evolves locally on the manifold (M, g). For instance, using the metric corresponding to a unit sphere S^2, an approximate solution may be obtained numerically by the EM method with $n = 2$. Two such sample trajectories $\hat{X}_t^1$ and $\hat{X}_t^2$ are shown in Figure 5.2. These two paths fairly match with the two simulated Brownian motions realized in the earlier exercise (see Figure 5.1) where the solution is obtained by solving SDE (5.1) and using exponential mapping at the end of each time step. The two solutions are superimposed over each other in Figure (5.2b). Thus, Equation (5.9) is the stochastically developed SDE corresponding to the SDE (5.1) when it is generalized from $\mathbb{R}^2$ to $\mathbb{R}^n$. The developed SDE is characterized by the presence of a non-zero drift term compared to its Euclidean counterpart, the SDE (5.1).

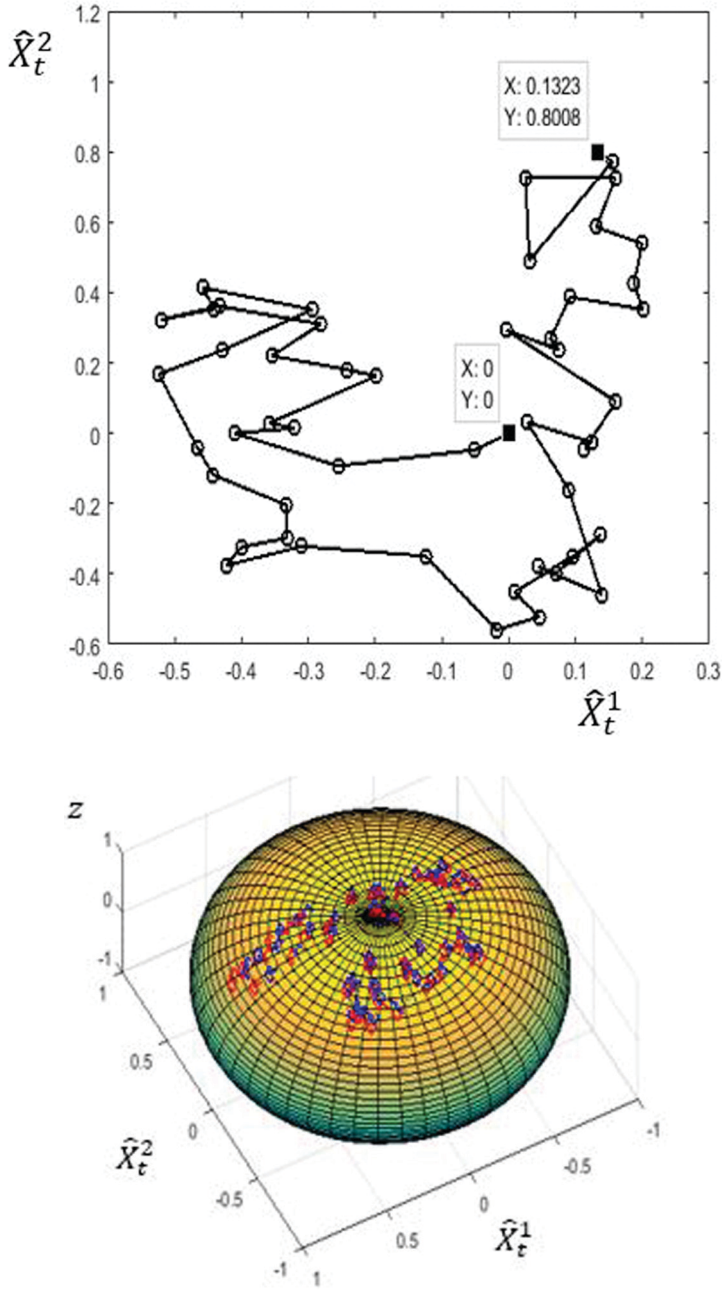


FIGURE 5.2 Sample solution by EM method of the stochastically developed SDE (5.9) with $n = 2$, while using the metric corresponding to a unit sphere manifold S^2 : (a) a 2-D plot of the solution $\hat{X}_t^1$ and $\hat{X}_t^2$ over the interval (0–0.5 s) with $\Delta t = 0.01$ and (b) the solution path on the sphere S^2 (shown in light squares); also see the solution (in dark squares) obtained from SDE (5.1) and transferred to the sphere manifold S^2 by using exponential mapping at each time.

5.2.1 A General Framework for Stochastic Development on a Manifold

We describe in this section an approach for stochastic development on a manifold by which it is possible to directly develop flows of stochastic dynamical systems posed as Ito SDEs via a suitably constructed metric (Hsu 2002, Sommer and Svane 2017, Khnel 2018, Mamajiwala and Roy 2022). Closely following Hsu (2002), we combine the basics of stochastic calculus with differential geometry to intrinsically recast an SDE, originally posed in a d-dimensional Euclidean space $\mathbb{R}^d$, on a Riemannian manifold M of the same dimension. The intrinsic nature of this approach implies that we do not need to embed M in a higher dimensional Euclidean space.

To this end, the canonical d-dimensional Euclidean basis† is related to a basis of the tangent plane T_xM at the point $x \in M$ with an additional construct of a $d + d^2$-dimensional manifold, called the frame bundle and denoted by FM.

While the d-dimensional component of FM is the base manifold M itself, the remaining d^2-dimensional part corresponds to linear transformations applied to vectors on T_xM. However, since the orthogonality of vectors is preserved for a Riemannian manifold, it would simplify matters to start with a set of orthonormal basis vectors and restrict attention to only orthogonal linear transformations of these basis vectors. In such a case, the dimension of FM is just $d + d = 2d$. Here, we may understand a frame bundle FM (Figure 5.3) as a space of pairs (x,u) with $x \in M$ and an isomorphism $u : \mathbb{R}^d \to T_x(M)$ defining a transformation of bases on the two tangent spaces. If one has $e_1, e_2, \ldots, e_d$ as the canonical basis vectors of $\mathbb{R}^d$, a basis frame on $T_x(M)$ is given by $u(e_i)$ or denoted simply by $ue_i, i = 1,2,\ldots,d.$

Denoting by FM_x the set of all basis vectors in $T_x(M)$, the elements of FM_x may, in general, be acted upon by $GL(d;\mathbb{R})$, the general linear group.‡ However, as already noted, we have presently restricted this action only to the special orthogonal group $SO(d;\mathbb{R})$. This means that any orthogonal linear transformation of FM_x is also a valid frame at x. FM_x is also called a fibre at x and the frame bundle may be thought of as the union of sets of fibres at different points on the manifold, i.e. $FM = \bigcup_{x \in M} FM_x$.

Roughly speaking, a fibre FM_x at a point x on M is defined as the space of frames attached to that point. A schematic visual representation of the frame bundle is shown

† *canonical basis*
A canonical basis is the natural basis of a coordinate space such as $\mathbb{R}^n$. In the *n*-dimensional Euclidean space, the standard basis consists of n distinct vectors forming the canonical basis and the vectors are $\{e_j : 1 \le j \le n\}$ where e_j denotes the vector with unity in the j^{ih} coordinate and zeros elsewhere.

‡ *general linear group*
The group of linear isomorphisms of $\mathbb{R}^n$ to $\mathbb{R}^n$, denoted by *GL* (*n*, *R*) is called the general linear group and is represented by $n \times n$ matrices of real elements. This is an open subset of $\mathbb{R}^{n^2}$, and so a manifold of dimension n^2.

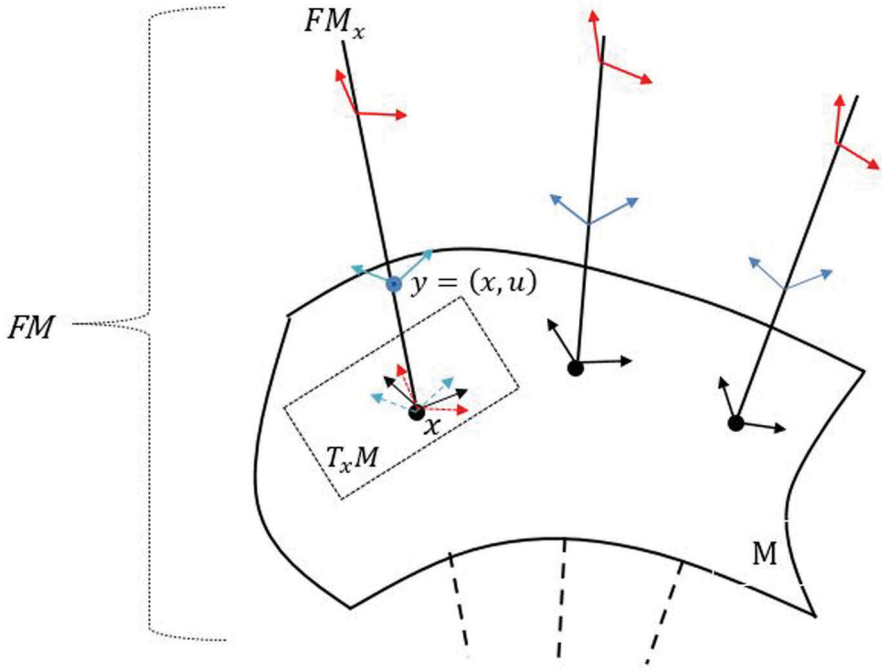


FIGURE 5.3 Frame bundle FM as the union of frames FM_x; each frame FM_x is the set of all basis vectors of $T_xM, x \in M$; the illustration is for the two-dimensional case.

in Figure 5.3 for $n = 2$. We may now define a surjective or onto map π: $FM \to M$ such that $\pi(x,u) = x$. Since FM itself is a (differentiable) manifold, the projection map π is also smooth.

By this additional construct of the frame bundle, we find that a frame at a point $x \in M$ provides us with a linear isomorphism between the Euclidean space $\mathbb{R}^d$ where the solution of a standard SDE evolves and the d-dimensional tangent plane $T_x(M)$ to M on which the solution needs to be projected. It is through the frame bundle that we can track these paths on M once we know how it evolves in $\mathbb{R}^d$. Following this objective, we proceed to consider the tangent space $T_y(FM)$ at $y \in FM$ with $y = (x, u)$ and understand how the space may be decomposed into vertical and horizontal subspaces, respectively denoted by V_yFM and H_yFM.

$T_y(FM)$ is presently a vector space of dimension $d+d$. We refer to a tangent vector $Y \in T_y(FM)$ as vertical if Y is tangent to the frame $FM_{\pi y = x}$. These vertical tangent vectors form a subspace $V_yF(M)$ of $T_y(FM)$ and it is of dimension d. $V_yF(M)$ signifies the changes in the basis at the base point $x = \pi y$ on M.

To define the horizontal subspace $H_y FM$, we consider a curve γ_t in FM which is basically a smoothly varying field of frames at points on the projected curve $\psi_t = \pi\gamma_t$ on M. A tangent vector $Y \in T_y(FM)$ is horizontal if it is tangential to a horizontal curve from y. A curve γ_t in FM starting from y is horizontal if, for each basis $e = \{e_1, e_2, \ldots, e_d\} \in \mathbb{R}^d$, the vector fields $\{\gamma_t(e_i)\}, i = 1,2,\ldots,d$ in M are parallel along ψ_t for each i.

With M equipped with a Riemannian connection ∇, we recall here (Section 4.2.6, Chapter 4) that a vector field V along the curve ψ_t on M is called parallel along ψ_t if $\nabla_{\dot{\psi}} V = 0$ for all t. That is, the vector V_{ψ_t} at the point ψ_t on M is the parallel transport of the vector V_{ψ_0} at ψ_0. $H_y FM$ is the space of such horizontal vectors $Y \in T_y FM$ lifted to the frame bundle. Thus, we have the direct-sum decomposition:

$$T_y FM = V_y FM \oplus H_y FM \tag{5.10}$$

Note that the projection $\pi : FM \to M$ induces an isomorphism $\pi_* : H_y FM \to T_{x=\pi y} M$ which defines a push-forward operation from $H_y FM$ to $T_x M$. See Section 4.2.1, Chapter 4, for a definition of push forward operation of tangent vectors from one manifold to another. Specifically, consider any vector $X \in T_x M$. The horizontal lift of X is then a unique horizontal vector $X^* \in H_y FM$ such that its projection returns the original vector itself, i.e. $\pi_* X^* = X$. Given the unit (orthonormal) coordinate basis vectors $e = (e_1, e_2, \ldots, e_d)$ in $\mathbb{R}_d$, the vector field H_e on FM at $y \in FM$ defined by $H_e(y) = (ye)^*$ gives the horizontal field on FM. Note that $H_e(y)$ is the horizontal lift of ye to y. $H_i := H_{e_i}; i = 1,2,\ldots,d$, are the associated horizontal vector fields of the frame bundle that span the horizontal subspace $H_y FM$ at each $y \in FM$. Figure 5.4 shows a curve γ_t on FM which is the horizontal lift of the curve ψ_t on M. The curve γ_t is horizontal in the sense that the tangent vector $\dot{\gamma}_t \ \forall t$ is horizontal and lies in $H_{\psi_t} FM$. As the figure shows, the vector fields $\pi_*(\dot{\gamma}_t)$ for different t are parallelly transported along ψ_t on M and move from one tangent space to another space with $\nabla_{\dot{\psi}}(\pi_*(\dot{\gamma}_t)) = 0$.

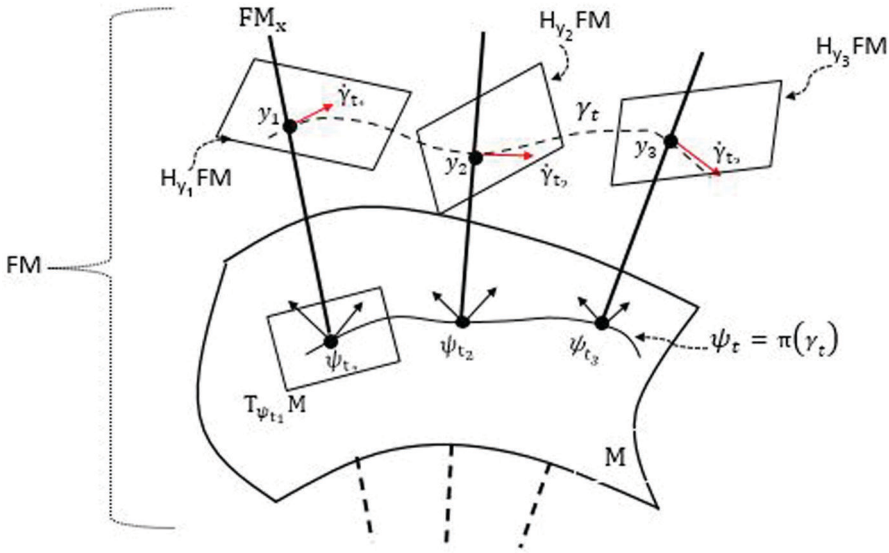


FIGURE 5.4 Horizontal lift γ_t on FM of the curve ψ_t in M – a two-dimensional case; $y_1 = \gamma_{t_1}$, $y_2 = \gamma_{t_2}$ and $y_3 = \gamma_{t_3}$, $H_{y_i} FM$, $i = 1,2,3$ are the spaces of horizontal vectors at typical points of the curve γ_t on the frame bundle.

5.2.1.1 Development on a Manifold of a Curve in $\mathbb{R}^d$

Keeping in view our requirement of developing on a manifold a known dynamic motion in the Euclidean space, we consider a curve χ_t in $\mathbb{R}^d$. Now, referring to the definition of γ_t, we have $\gamma_t^{-1}\dot{\psi}_t \in \mathbb{R}^d$ and this helps in having the curve χ_t from the following equation:

$$\chi_t = \int_0^t \gamma_s^{-1}\dot{\psi}_s ds \tag{5.11}$$

χ_t may now be considered as the anti-development of ψ_t. Note that Equation (5.11) leads to the following ODE:

$$\gamma_t \dot{\chi}_t = \dot{\psi}_t \tag{5.12}$$

By the definition of horizontal vector fields, we also have:

$$H_{\dot{\chi}_t}(\gamma_t) = (\gamma_t \dot{\chi}_t)^* = \dot{\psi}(t)^* = \dot{\gamma}_t \tag{5.13a}$$

and

$$H_{\dot{\chi}_t}\left(\gamma_t\right)=\left(\gamma_t\dot{\chi}_t\right)^*=\left(\gamma_t e_i\right)^*\dot{\chi}_t^i=H_i\left(\gamma_t\right)\dot{\chi}_t^i \tag{5.13b}$$

Hence, we have:

$$\dot{\gamma}_t=H_i\left(\gamma_t\right)\dot{\chi}_t \tag{5.14}$$

Thus, the anti-development χ_t and the horizontal lift γ_t of a curve ψ_t on M are simply related by an ODE. Here, we realize that if we start with an Euclidean curve χ_t in $\mathbb{R}^d$ and a frame γ_0 at the point x_0 on M, the unique solution of the above ODE gives the horizontal curve γ_t in FM. We refer to this horizontal curve as the development of χ_t in the frame manifold FM. Its projection on M given by $\psi_t=\pi\gamma_t$ is the very development what we are interested in, i.e. of transferring the curve χ_t in $\mathbb{R}^d$ to M. We notice that solution of Equation (5.14) requires knowledge of the operator $H_i(.)$. This is addressed in the following.

Suppose that we adopt any local chart $x=\left\{x^i\right\}, i=1,2\ldots,d$ on a neighbourhood $U\in M$. By the inverse of the projection map π, this local chart on the base manifold M induces a local chart on $\hat{U}=\pi^{-1}\left(U\right)$ in FM. Thus, letting $X_i=\dfrac{\partial}{\partial x^i}$, $i=1,2,\ldots,d$ as the coordinate basis vectors in T_xM, we have, for a frame $q\in\hat{U}$, $qe_{\mathrm{i}}=Q_i^j\,X_j$ for some matrix $Q=\left[Q_i^j\right]\in SO\left(d,\mathbb{R}\right)$. Recall that if we start with an orthonormal frame, horizontal parallel transport for a Riemannian manifold must retain the orthonormality. Hence, we take Q to be orthonormal. In general, we get $\left(x,q\right)\in\mathbb{R}^{d+d}$ as the local chart for $\hat{U}$. Then, the vertical subspace V_qFM is spanned by $X_{ij}=\dfrac{\partial}{\partial Q_j^i}$, $1\le i,j\le d$. Also, the vector fields $\left\{X_i,\,X_{ij}\right\}$, $1\le i,j\le d$ span T_qFM. An expression for the horizontal vector field H_i in terms of the local coordinates is given [Hsu 2002] as follows:

$$H_i\left(q\right)=Q_i^jX_j-Q_i^jQ_m^l\Gamma_{jl}^k\left(x\right)X_{km} \tag{5.15}$$

Instead of the curve χ_t being deterministic, suppose that it is a stochastic process and specifically it is governed by an SDE in $\mathbb{R}^d$. One may follow the procedural steps similar to those enumerated in the deterministic case above so as to get the corresponding stochastically developed SDE on a manifold. This is indeed the basic concept of the geometric optimization strategy, i.e. to stochastically develop the Langevin SDE (4.103) of $\mathbb{R}^d$ on a Riemannian manifold of the same dimension and to subsequently solve the developed SDE for the optimal solution.

5.2.2 Stochastic Development of an SDE on a Manifold

The case of a pure Brownian motion, i.e., $d\chi_t = dB_t$ in $\mathbb{R}^d$ and its development on a manifold is detailed in Hsu [2002]. We consider here a diffusion process $\chi_t \in \mathbb{R}^d$ governed by a general SDE with a non-zero drift term:

$$d\chi_t^i = \alpha^i\left(\chi_t\right)dt + \sigma_j^i\left(\chi_t\right)dB_t^j, i = 1,2,\ldots,d \text{ and } j = 1,2,\ldots,n \tag{5.16}$$

Note that, in general, the drift and diffusion terms on the RHS of the SDE may be functions of time *t* also. Development of the SDE (5.16) requires an extension of Equation (5.14) to the stochastic case. A familiar route is to write this equation in the Stratonovich sense,[§] i.e.:

$$\dot{\gamma}_t = H_i\left(\dot{\gamma}_t\right) o \dot{\chi}_t \Rightarrow d\gamma_t = H_i\left(\dot{\gamma}_t\right) o d\chi_t \tag{5.17}$$

[§] Stratonovich sense

An SDE governing a stochastic process X_t is usually written in either Stratonovich or Ito sense:

$$dX_t = \alpha\left(X_t\right)dt + \sigma\left(X_t\right) \circ dB_t \left(\text{Stratonovich}\right) \tag{i}$$

$$dX_t = \alpha\left(X_t\right)dt + \sigma\left(X_t\right)dB_t \left(\text{Ito}\right) \tag{ii}$$

In the integral form, solutions of SDEs involve integrals of the type $\int_0^t \sigma\left(X_s\right)dB(s)$. That is:

$$X_t = \int_0^t \alpha\left(X_s\right)ds + (S)\int_0^t \sigma\left(X_s\right)dB_s \left(\text{Stratonovich}\right) \tag{iii}$$

$$X_t = \int_0^t \alpha\left(X_s\right)ds + (I)\int_0^t \sigma\left(X_s\right)dB_s \left(\text{Ito}\right) \tag{iv}$$

$\int_0^t \sigma\left(X_s\right)dB_s$ is known as a stochastic integral wherein both the integrand and the integrator involve a stochastic process. (S) and (I) written before the integral respectively denote the Stratonovich and Ito type of stochastic integral. The integral is not the classical one in Stieltjes sense since Brownian motion B_t is not differentiable and its total variation not finite (see Appendix 4). The integral if approximated in the form of a Riemannian sum by the following limiting sequence, takes the form:

$$\mathfrak{T} = \lim_{N\to\infty} \sum_{j=0}^{N-1} \sigma\left(X_{t^*}\right)\left(B\left(t_{j+1}\right) - B(t_j)\right) \tag{v}$$

Here, we use a partition Π_N of the interval $[0,t]$ given by $0 = t_0 < t_1 < \ldots < t_N = t$ and with $\Delta_N = \max_{0\le j\le N-1}\left(t_{j+1} - t_j\right)$. In principle, one can create an infinite sequence of such summands as in (v) corresponding to a choice of $t^* \in \left[t_j, t_{j+1}\right] \forall j$ and thus define an approximation to $\mathfrak{T}$ as $\mathfrak{T}' = \sum_{j=0}^{N-1} \sigma\left(X_{t^*}\right)\left(B_{t_{j+1}} - B_{t_j}\right)$. Thus, the choice of t^* matters in defining a stochastic integral. Specifically, the integral for $t^* = t_j$ is called the Ito integral [Ito 1951]. If $t^* = \left(t_j + t_{j+1}\right)/2$ (the

mid point in $\left[t_j, t_{j+1}\right]$), it leads to another integral representation known as the Stratonovich integral [Stratonovich, 1966]. To write the Stratonovich SDE in an equivalent Ito form, we take the stochastic integral in the Stratonovich sense and perform the following manipulations.

$$(S)\int_0^t \sigma(X_s)\,dB_s = \int_0^t \sigma\left(\frac{1}{2}\left(X_{s+\Delta s} + X_s\right)\right)dB_s \tag{vi}$$

Writing $\sigma\left(\frac{1}{2}\left(X_{s+\Delta s} + X_s\right)\right) = \sigma\left(X_s + \frac{1}{2}\left(X_{s+\Delta s} - X_s\right)\right)$ and approximating it by Taylor expansion up to first-order terms in $\Delta X_s = X_{s+\Delta s} - X_s$, one has:

$$(S)\int_0^t \sigma(X_s)\,dB_s = (I)\int_0^t \sigma(X_s)\,dB_s + \frac{1}{2}\int_0^t \frac{d\sigma}{dx}\Delta X_s\,dB_s \tag{vii}$$

Substituting the expression on the RHS of (ii) for ΔX_i in the last equation, one gets:

$$(S)\int_0^t \sigma(X_s)\,dB_s = (I)\int_0^t \sigma(X_s)\,dB_s + \frac{1}{2}\int_0^t \frac{d\sigma}{dx}\left(\alpha(X_s)\,ds + \sigma(X_s)\,dB_s\right)d\boldsymbol{B}_s$$

$$= (I)\int_0^t \sigma(X_s)\,dB_s + \frac{1}{2}\int_0^t \frac{d\sigma}{dx}\sigma(X_s)\,ds \tag{viii}$$

The last result in (viii) is obtained by using the rule that $dB_s\,ds = 0$ and $dB_s\,dB_s = ds$. The equivalent Ito SDE is then signified by the presence of an additional drift term as:

$$dX_t = \left(\alpha(X_t) + \frac{1}{2}\frac{d\sigma}{dx}\sigma(X_t)\right)dt + \sigma(X_t)\,dB_t \tag{ix}$$

Note that when $\sigma(X_t)$, the diffusion coefficient is not dependent on the system state X_t, the system is referred to as one with an additive noise and when it depends on the state X_t, the noise is called multiplicative. It is obvious that in the former case, one finds no disparity between the two approaches (since $\frac{d\sigma}{dx} = 0$ in (ix) above). The equivalent Ito SDE may also be rewritten as:

$$dX_t = \alpha(X_t)\,dt + \sigma(X_t)\,dB_t + \frac{1}{2}d\left[\sigma(X_t), B\right] \tag{x}$$

$[.,.]$ stands for the quadratic covariation between two stochastic processes (Roy and Rao 2017). By Ito's formula (Appendix 4) applied to the function $\sigma(X_t)$ and using the definition of the quadratic covariation, one obtains the equivalent form in (x). In the case of Ito integral, the solution X_t retains the Markovian and martingale properties (van Kampen 1981, Moon and Wettlaufer 2014, Roy and Rao 2017). This may help in utilizing the available theory of martingales that provides a computational tool of considerable benefit; specific applications may be in mathematical finance, stochastic control and optimization. But it requires new rules of calculus namely Ito Calculus (Roy and Roy 2017). On the other hand, the Stratnovich approach lacks the Markovian property but finds advantage in following the same rules of classical calculus. It amounts to interpreting the white noise process as a regular derivative of Brownian motion, even though an ideal white noise does not exist. The approach is frequently used in engineering and physical sciences.

From Equation (5.15), the horizontal vector field H_i at γ_t is locally given by:

$$H_i\gamma_t = Q_i^j X_j - Q_i^j Q_m^l \Gamma_{jl}^k X_{km} \tag{5.18}$$

$$H_i\gamma_t = Q_i^j X_j - Q_i^j Q_m^l \Gamma_{jl}^k X_{km} \tag{5.18}$$

In the Stratonovich sense, Equation (5.17) for $\gamma(t) = \left\{x_t^i, Q_j^i(t)\right\}$ may be written as:

$$dx_t^i = Q_j^i(t) \circ d\chi_t^j$$

$$dQ_j^i(t) = -\Gamma_{kl}^i Q_j^l Q_m^k \circ d\chi_t^m \tag{5.19a,b}$$

In the Ito sense, Equation (5.19a) is:

$$dx_t^i = Q_j^i(t) d\chi_t^j + \frac{1}{2} d\left[Q_j^i(t), d\chi_t^j\right] \text{ (here [., .] indicates quadratic covariation)}$$

$$= Q_j^i(t)\alpha^j(\chi_t)dt + Q_j^i(t)\sigma_m^j(\chi_t)dB_t^m + \frac{1}{2} d\left[Q_j^i(t), d\chi_t^j\right] \tag{5.20}$$

Here [.,.] indicates quadratic covariation (Roy and Rao 2017) between two stochastic processes which is similar to the quadratic variation defined in Appendix 4.

$$dQ_j^i(t) = -\Gamma_{kl}^i(x_t) Q_j^l(t) Q_m^k(t)\, d\chi_t^m + \frac{1}{2} d\left[-\Gamma_{kl}^i(x_t) Q_j^l(t) Q_m^k(t), \chi_t^j\right] \tag{5.21a}$$

Since the expression for $\Gamma_{kl}^i(x_t)$ is deterministic, the last term on the RHS of the last equation vanishes and therefore:

$$dQ_j^i(t) = -\Gamma_{kl}^i(x_t) Q_j^l(t) Q_m^k(t)\, d\chi_t^m \tag{5.21b}$$

Note that the 2nd term on the RHS of Equation (5.20) is the martingale part** dM_t^i:

** *martingale part*
The SDE (5.16), if written in the integral form, shows that the second term on the RHS corresponds to an Ito integral of the type:

$$\mathfrak{T}_t = \int_0^t Y_s dB_s. \tag{i}$$

$$dM_t^i = Q_j^i(t)\,\sigma_m^j(\chi_t)\,dB_t^m \tag{5.22}$$

By letting $\vartheta = Q\sigma$, we have:

$$dM_t^i = \vartheta_{im}\,dB_t^m \tag{5.23}$$

Therefore:

$$\begin{aligned} d\left[M_t^i, M_t^j\right] &= \left[dM_t^i, dM_t^j\right] \\ &= \vartheta_{im}\,\vartheta_{jn}\left[dB_t^m, dB_t^n\right] \\ &= \left[\vartheta^T\vartheta\right]_{ij} dt \left(\text{here square brackets indicate a matrix}\right) \end{aligned} \tag{5.24}$$

The last step is obtained since the quadratic variation of Brownian motion given by $[B,B](t) = t$ (Equation 4.14 of Chapter 4) and hence

---

Let $\{\mathcal{F}_t\}$ be the filtration generated by Brownian motion B_t up to time t and let $Y_t \in \mathcal{F}_t$ be an adapted stochastic process. Now, consider the sequence $Y^{(N)} = \sum_{j=0}^{N-1} Y_{t_j} I_{t_j, t_{j+1}}(t)$ where $I_{t_j, t_{j+1}}(t) = 1$ if $t_j \le t \le t_{j+1}$ and zero otherwise. Let $\mathfrak{T}_t^{(N)} = \int_0^t Y^{(N)} dB_s$. Since Brownian motion is continuous, $\mathfrak{T}_t^{(N)}$ is continuous for all N and $\mathfrak{T}_t = \lim_{N\to\infty} \mathfrak{T}_t^{(N)}$. Having a partition Π_N of the interval $[0,t]$ and with $t' < t$, one has:

$$\begin{aligned} E\left[\mathfrak{T}_t^{(N)} \mid \mathcal{F}_{t'}\right] &= E\left[\left(\int_0^{t'} Y^{(N)} dB_s + \int_{t'}^{t} Y^{(N)} dB_s\right)\mathcal{F}_{t'} \mid\right] \\ &= \int_0^{t'} Y^{(N)} dB_s + E\left[\sum_{t' \le t_j^{(N)} \le t_{j+1}^{(N)} \le t} Y_j^{(N)} \Delta B_j \mid \mathcal{F}_{t'}\right] \\ &= \int_0^{t'} Y^{(N)} dB_s + \left[\sum_{t' \le t_j^{(N)} \le t_{j+1}^{(N)} \le t} E\left[Y_j^{(N)} E\left[\Delta B_j \mid \mathcal{F}_{t_j^{(N)}}\right] \mid \mathcal{F}_{t'}\right]\right] \\ &= \int_0^{t'} Y^{(N)} dB_s \left(\text{since } \Delta B_j \sim \mathcal{N}\left(0, \sqrt{\Delta t_j}\right) \Delta E\left[\Delta B_j \mid \mathcal{F}_{t_j^{(N)}}\right] = 0\right) \\ &= \mathfrak{T}_{t'}^{(N)} \end{aligned} \tag{ii}$$

Hence $\mathfrak{T}_t^{(N)}$ is a martingale and so is $\mathfrak{T}_t$. Thus, the second term on the RHS of Equation (5.20) is a martingale.

$\left[dB_t^m, dB_t^n\right] = d\left[B_t^m, B_t^n\right] = dt$, if $m = n$. Now, with a metric g on the manifold M, we have for a frame $qe_l = Q_l^j\ X_j$ and $qe_l, qe_{mg} = \delta_{lm}$ where δ_{lm} is the Kronecker delta (see also Equation 4.16). This may be stated as:

$$\begin{aligned}
\delta_{lm} &= \left\langle qe_l,\ qe_m \right\rangle_g \\
&= \left\langle Q_l^p\ X_p,\ Q_m^q\ X_q \right\rangle_g \\
&= Q_l^p Q_m^q \left\langle X_p, X_q \right\rangle_g \\
&= Q_l^p Q_m^q \left\langle \frac{\partial}{\partial x^p}, \frac{\partial}{\partial x^q} \right\rangle_g \\
&= Q_l^p Q_m^q \mathfrak{g}_{pq} \\
\Rightarrow QQ^T \mathfrak{g} &= I \\
\Rightarrow QQ^T &= \mathfrak{g}^{-1} = Q^T Q \left(\mathfrak{g} \text{ is symmetric}\right) \qquad (5.25)
\end{aligned}$$

Referring to Equation (5.24), we get:

$$\begin{aligned}
d\left\langle M_t^i, M_t^j \right\rangle &= \left[\vartheta^T \vartheta\right]_{ij} dt = \left[\sigma^T Q^T Q \sigma\right]_{ij} dt = \left[\sigma^T \mathfrak{g}^{-1} \sigma\right]_{ij} dt \\
\Rightarrow \vartheta^T \vartheta &= \sigma^T \mathfrak{g}^{-1} \sigma = \sigma^T \sqrt{\mathfrak{g}^{-1}}^T\ \sqrt{\mathfrak{g}^{-1}}\ \sigma \\
\Rightarrow \vartheta &= \sqrt{\mathfrak{g}^{-1}}\,\sigma \qquad (5.26)
\end{aligned}$$

In view of Equation (5.21b), the last term in the RHS of Equation (5.20) becomes:

$$\begin{aligned}
d\left[Q_j^i(t), d\chi_t^j\right] &= d\left[-\Gamma_{kl}^i\left(x_t\right) Q_j^l(t) Q_m^k(t) d\chi_t^m, d\chi_t^j\right] \\
&= -\Gamma_{kl}^i\left(x_t\right) Q_j^l(t) Q_m^k(t) d\left[d\chi_t^m, d\chi_t^j\right] \\
&= -\Gamma_{kl}^i\left(x_t\right) Q_j^l(t) Q_m^k(t) \left[\alpha^m\left(\chi_t\right) dt + \sigma_p^m\left(\chi_t\right) dB_t^p, \alpha^j\left(\chi_t\right) dt + \sigma_q^j\left(\chi_t\right) dB_t^q\right] \\
&= -\Gamma_{kl}^i\left(x_t\right) Q_j^l(t) Q_m^k(t) \sigma_p^m\left(\chi_t\right) \sigma_q^j\left(\chi_t\right) \left[dB_t^p,\ dB_t^q\right]
\end{aligned}$$

$$= -\Gamma^i_{kl}\left(x_t\right) Q^l_j(t) Q^k_m(t)\, \sigma^m_p\left(\chi_t\right) \sigma^j_p\left(\chi_t\right) dt$$

$$= -\Gamma^i_{kl}\left(x_t\right) \left[Q^l_j(t)\, \sigma^j_p\left(\chi_t\right)\right] \left[Q^k_m(t)\, \sigma^m_p\left(\chi_t\right)\right] dt$$

$$= -\Gamma^i_{kl}\left(x_t\right) [Q\sigma]_{lp} [Q\sigma]_{kp}\, dtg$$

$$= -\Gamma^i_{kl}\left(x_t\right) \left[(Q\sigma)(Q\sigma)^T\right] dt$$

$$= -\Gamma^i_{kl}\left(x_t\right) \left[\sigma Q Q^T \sigma^T\right] dt$$

$$= -\Gamma^i_{kl}\left(x_t\right) \left[\sigma \mathfrak{g}^{-1} \sigma^T\right] dt \quad \left(\text{from Equation } 5.25\right) \tag{5.27}$$

In view of Equations (5.25), (5.26) and (5.27), we rewrite Equation (5.20) as:

$$dx^i_t = \left[\sqrt{\mathfrak{g}^{-1}}\right]_{ij} \pm^j dt + \left[\sqrt{\mathfrak{g}^{-1}}\sigma\right]_{im} dB^m_t - \frac{1}{2}\Gamma^i_{kl}\left(x_t\right) \left[\sigma \mathfrak{g}^{-1} \sigma^T\right]_{kl} dt$$

$$= \left(\left[\sqrt{\mathfrak{g}^{-1}}\right]_{ij} \alpha^j - \frac{1}{2}\Gamma^i_{kl}\left(x_t\right) \left[\sigma \mathfrak{g}^{-1} \sigma^T\right]_{kl}\right) dt + \left[\sqrt{\mathfrak{g}^{-1}}\sigma\right]_{im} dB^m_t \tag{5.28}$$

$x^i_t, i = 1,2,\ldots,d$ gives the evolution of a point on the curve ψ_t which is the projection of the horizontal (curve) lift γ_t in FM on the base manifold M. Thus, Equation (5.28) is the stochastically developed SDE corresponding to the SDE (5.16) in $\mathbb{R}^d$. In the absence of the drift term $\alpha\left(\chi_t\right)$ and for additive noise with $\sigma = I_{d\times d}$, a unity matrix, Equation (5.28) reduces to Equation (5.9) which is the stochastically developed SDE corresponding to pure Brownian motion in $\mathbb{R}^d$ and which was earlier obtained by using Laplace-Beltrami operator.

Before we proceed to illustrate the application of stochastic development in function optimization problems, a word about anti-development. By anti-development, there exists a unique γ_t, the horizontal lift on the frame bundle FM of a smooth curve ψ_t on M as a solution of an ODE (Kobayashi and Nomizu 1963). In turn this horizontal lift corresponds to a unique curve χ_t in the Euclidean space. For instance, in human activity recognition (Yi *et al*. 2011), one transfers the collected manifold data to the Euclidean space by anti-development and performs stochastic modelling by traditional means in the flat space.

5.3 NON-CONVEX FUNCTION OPTIMIZATION BASED ON STOCHASTIC DEVELOPMENT

In this section, we consider the application of stochastic development to an optimization problem that involves a non-convex objective function. To this end, we again refer to the analogy between statistical sampling and optimization (Section 4.5, Chapter 4). We have already utilized this analogy in Chapter 4 to pose the optimization problem as the solution of an overdamped Langevin SDE (see Equation 4.103) in $\mathbb{R}^d$ and to transfer the solution to the manifold by exponential mapping at each time step. Here, we stochastically develop the SDE and solve directly on the manifold within a stochastic search framework. We name this scheme as Geometrically Adapted Langevin algorithm (GALA) as already mentioned in the introduction of the chapter. Note that a strictly positive, smooth, scalar-valued and n-dimensional non-convex objective function $f(x): \mathbb{R}^n \to \mathbb{R}$ may be looked upon, at least locally, as an energy-like functional in the space of the design variables. Using this functional to represent a Riemannian manifold, we derive the associated metric g and the connection to stochastically develop the Langevin SDE (4.103). During stochastic search involving a non-convex function, the matrix $\mathfrak{g}$ associated with the metric g (both may be loosely called the metric) may sometimes become negative-definite, particularly during the initial stages. As such, an additive regularizer of the type $\Xi I_{d\times d}$ where $\Xi \in \mathbb{R}^+$ may be used to ensure the positive definiteness of $\mathfrak{g}$. The following examples on two test functions (Tang et al. 2009) show the efficacy of the stochastic development approach in handling higher dimensional optimization problems.

Example 5.1. We consider minimization of the same Ackley function as considered in Example 4.9 of Chapter 4. With $x \in \mathbb{R}^n$, the Ackley function is rewritten below:

$$f(x) = -a\exp\left(-b\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(cx_i)\right) + a + \exp(1) \tag{5.29}$$

where n is the dimension of the optimization problem.

Solution. The overdamped Langevin SDE (4.103) in $\mathbb{R}^d$ in differential form:

$$dX_t = -\beta_t \nabla f(X)\,dt + \sqrt{2\beta_t}\,dB_t \tag{5.30}$$

β_t is an annealing parameter (as used in the simulated annealing method of optimization). Note that β_t expedites a more exhaustive search of the search space during the initial stages. One gets the stochastically developed SDE corresponding to Equation (5.30) as:

$$dX_t = -\beta_t\left(\sqrt{\mathfrak{g}^{-1}}\,\nabla f(X) - \frac{1}{2}\mathfrak{g}^{-1}\Gamma\right)dt + \sqrt{2\beta_t\mathfrak{g}^{-1}}\,dB_t \tag{5.31}$$

Since we need to compute $\mathfrak{g}^{-1}$ and to arrive at the developed SDE (5.31), the present scheme GALA is not gradient-free. However, when the gradient of the objective function is available, the goal of optimization is obviously expedited by a relatively faster convergence. Details on computation of the matrix $\mathfrak{g}$ and the connection matrices corresponding to the Christoffel symbols are available in Appendix 4. Results are obtained for the function with dimensions $n = 10$ and 40 and an ensemble size of only five particles $N_p = 5$ is used. The parameter β_t is initially assumed to be 5E4. It is reduced according as $\beta_t^{k+1} = \beta_t^k / e^{0.01k}$ as iterations progress, where the superscript k indicates the k^{th} iteration. Figures 5.5 and 5.6 show these results for $n = 10$ and $n = 40$ respectively. Results are also obtained for $n = 40$ via the RMALA and classical MALA (Chapter 4) and are shown in Figure 5.7. Comparison of the results indicate that the last two approaches fail to converge for higher dimensional problems.

■

Example 5.2. Consider minimization of another test function – the Rastrigin function (Tang et al. 2009):

$$f(x) = \sum_{j=1}^{m} \left\{ x_j^{\,2} - 10\cos 2\pi x_j + 10 \right\} \tag{5.32}$$

Solution. The matrix $\mathfrak{g}$ and the connection matrices are detailed in Appendix 5. The optimization results obtained for $n = 40$ are shown in Figure 5.8. The parameter β_t is initially assumed to be 5E2 and as in the earlier example, it is reduced as iterations progress. An ensemble size of $N_p = 5$ is adopted during the computations.

■

The algorithm parameters are so chosen (by trial and error) as to represent the best performance of the scheme. As expected, the annealing parameter β_t expedites a more exhaustive search of the design space during the initial stages. As can be noticed from the results obtained for the two test functions in the last two examples, the stochastically developed version converges faster, i.e. with fewer steps. Stochastic development has applications in statistical analysis on manifolds associated with medical anatomy and deep learning numerics. An interested reader may find, for example, efforts at constructing regression models for manifold-valued nonlinear data by stochastic development in Khnel and Sommer (2017ü) and Kühnel *et al*. (2019).

5.3.1 Issues Related to Computation of '$\mathfrak{g}$' Matrix and Its Derivatives

Derivatives of the objective function are needed to evaluate $\mathfrak{g}$ and the matrices associated with Christoffel symbols. For large size dimensional problems, one may encounter issues in evaluating these derivatives in closed form; indeed they may not even exist everywhere for a class of objective functions. A better route to evaluate the metric and connection coefficients is perhaps through a numerical approach. In this

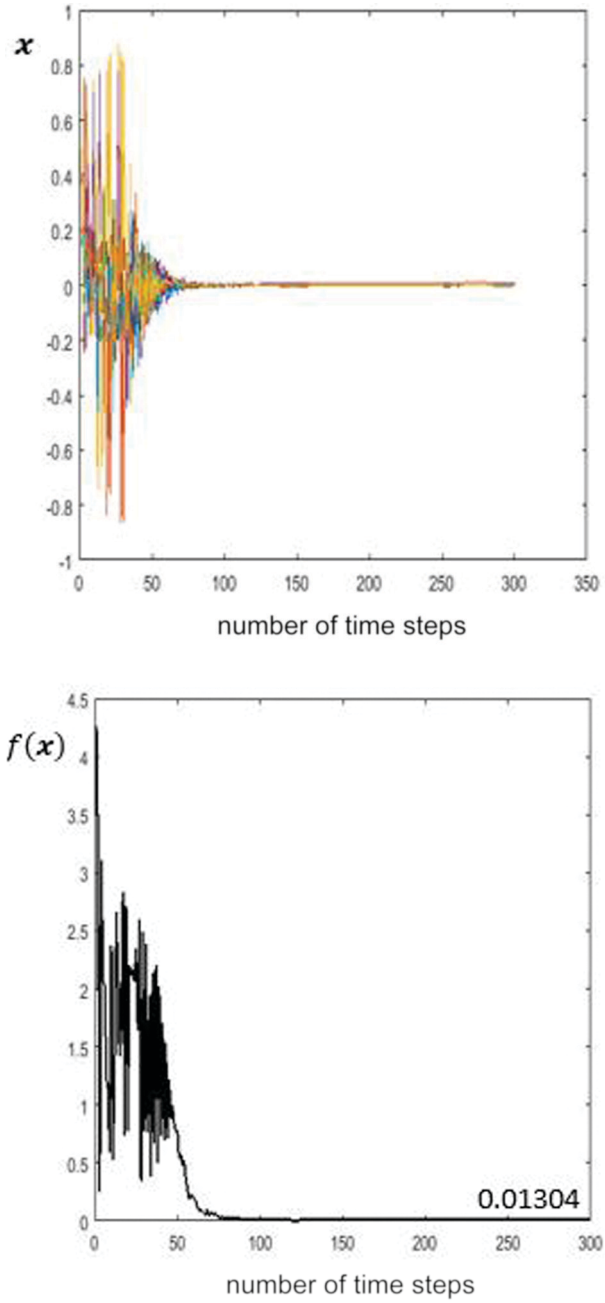


FIGURE 5.5 Optimization by GALA of 10-dimensional Ackley function ($n = 10$): (a) evolution of the solution x of the stochastically developed SDE (5.31) and (b) evolution of the objective function $f(x)$, $dt = 0.01$, $N_p = 5$.

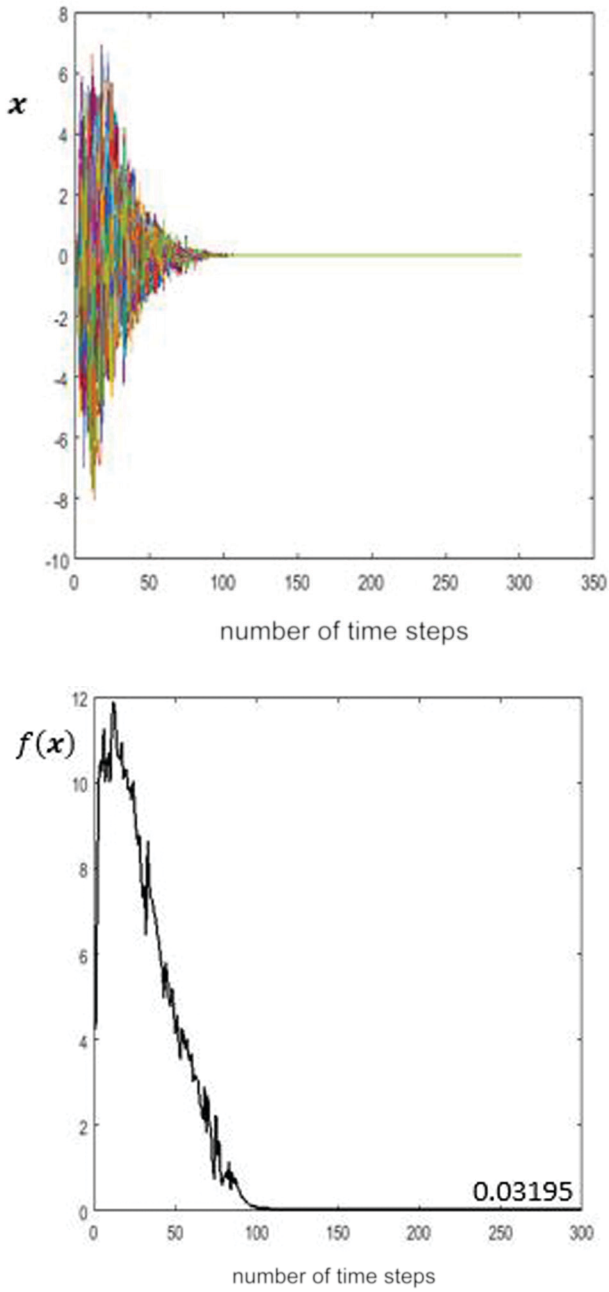


FIGURE 5.6 Optimization by GALA of 40-dimensional Ackley function ($n = 40$); (a) evolution of the solution x of the stochastically developed SDE (5.43) and (b) evolution of the objective function $f(x)$, $\Delta t = 0.5$, $N_p = 5$.

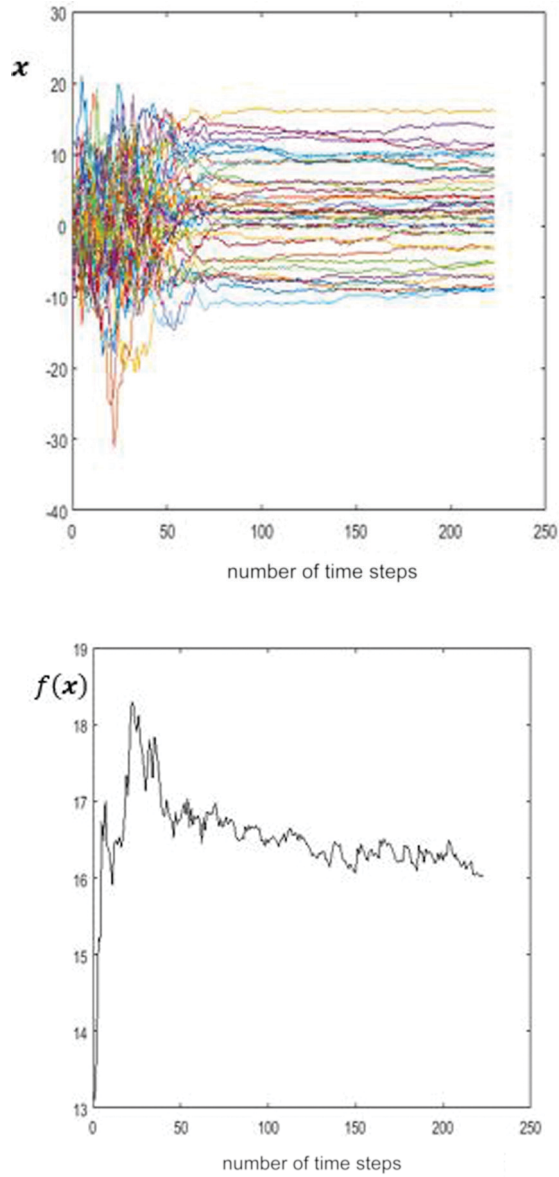
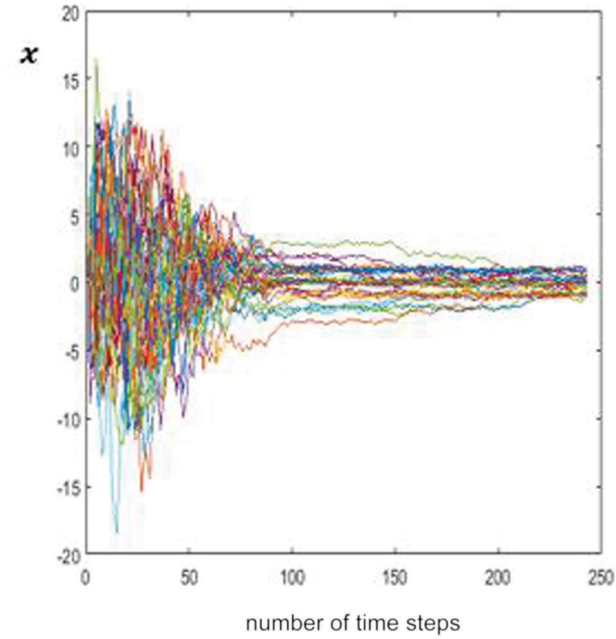


FIGURE 5.7a–b Optimization of 40-dimensional Ackley function (n = 40) by RMALA (Section 4.6.2, Chapter 4) using the exponential mapping step; (a) evolution of the solution $\boldsymbol{x}$ of the SDE (4.155) of Chapter 4 and (b) evolution of the objective function $f(\boldsymbol{x})$, $\Delta t = 0.01$, $Np = 5$. Optimization of 40-dimensional Ackley function ($n = 40$) by classical MALA (Equation 4.154) using steepest descent step; (c) evolution of the solution x and (d) evolution of the objective function $f(x)$, $\Delta t = 0.01$, $Np = 5$.



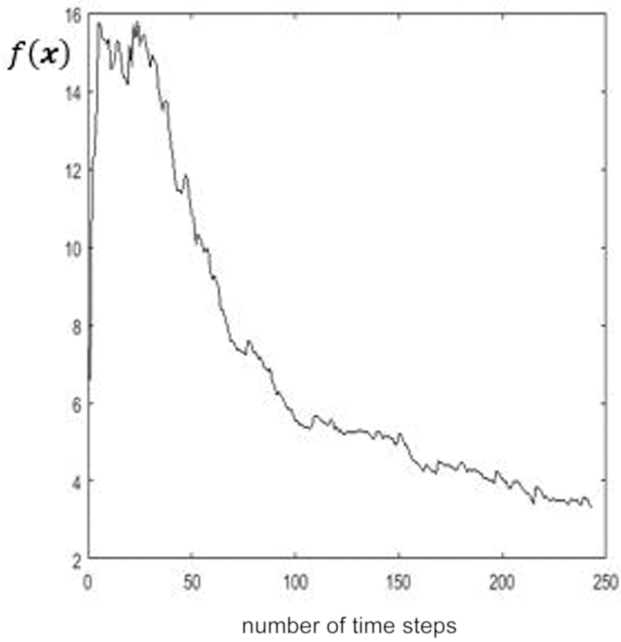


FIGURE 5.7c–d (Continued)

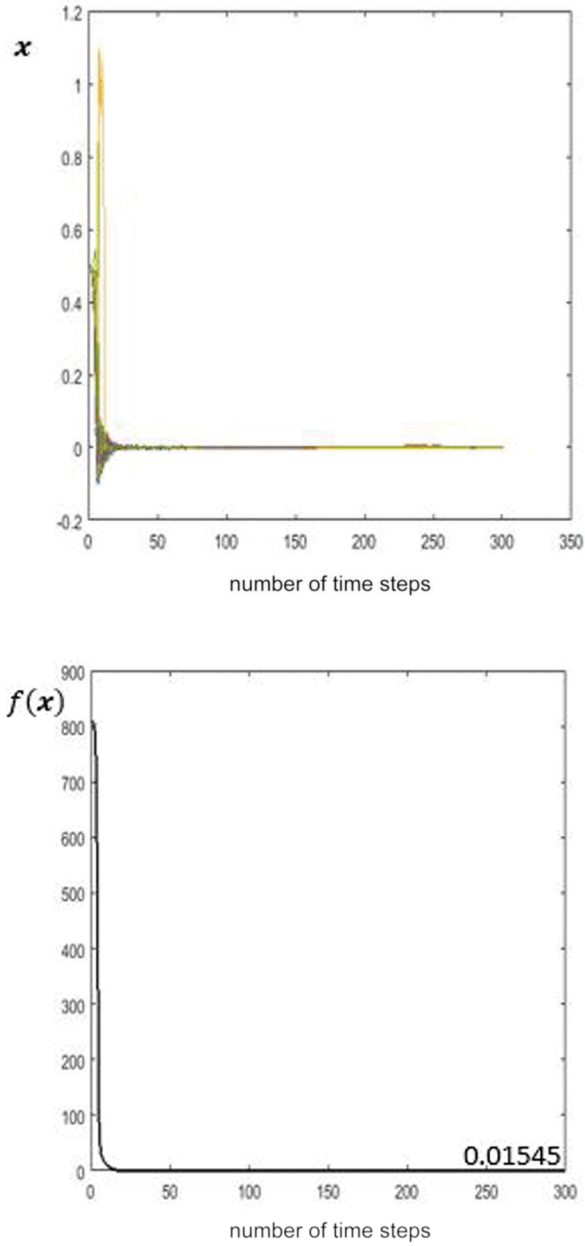


FIGURE 5.8 Optimization by GALA (with stochastic development) of 40-dimensional Rastrigin function ($n = 40$); (a) evolution of the solution x and (b) evolution of the objective function $f(x)$, $\Delta t = 0.01$, $Np = 5$.

regard, one elegant way to get these derivatives is to adopt the strategy of approximating functions (Powell 1981, Buhmann 2003, Shapiai *et al*. 2011) by using radial basis/ kernel functions. Use of kernel functions in nonlinear regression finds applications in pattern recognition (Shawe-Taylor and Christianini, 2004) and robotics (Das and Yip 2020). Bromhead and Lowe (1988) and Moody and Darken (1989) used radial basis functions in the design of neural networks for multivariable interpolation problems. A radial basis function (RBF) $\psi(.)$ is real valued and depends on the Euclidean distance from the origin or a specified centre. RBFs are a set of functions which have the same value at a fixed distance from the central point, being characterized by a monotonic decay with distance from the central point. A typical univariate radial function is the Gaussian RBF:

$$\psi(x) = e^{-\left(\frac{|x-c|}{r}\right)^2} \tag{5.33a}$$

Another popular choice is the bump function:

$$\psi(x) = e^{-\frac{1}{1-(|x-r|)^2}} \tag{5.33b}$$

In the present context, let us consider evaluating a multi-variable function $f(x): \mathbb{R}^n \to \mathbb{R}$ and its derivatives at any $x \in \mathbb{R}^n$ using an RBF. For instance, one may obtain $f(\boldsymbol{x})$ as:

$$f(x) = E_P\left[f(y)\,\psi(|y-x|)\right] \tag{5.34}$$

where $E[.]$ is the expectation operator with respect to a probability measure p associated with the vector RV y. $\psi(.)$ is the RBF from $C^{\infty}(\mathbb{R})$ and is compactly supported. Note that the functions f appearing on the left and right hand sides of Equation (5.46) are not identical. Indeed, while $f(x)$ on the LHS could be considered as a regularized (smoothened) approximation for the original function, we presently overlook this distinction to simplify the notations. The argument $|y-x|$ is indicative of the Euclidean distance of y from the fixed point x. If we consider an infinitesimal ball $B(x,\delta x)$ centred at x with a radius δx and obtain realizations $y_j, j = 1,2,\ldots N$ restricted to this hypersphere, the expectation operation in Equation (5.46) approximates to:

$$f(x) = \frac{1}{N}\sum_j f(y_j)\,\psi(|y_j - x|) \tag{5.35}$$

One may conveniently adopt a uniform measure for the variable y over the ball $B(x,\delta x)$ and thus obtain $f(x)$ by the averaging operation in the last equation. Since

a uniform distribution $U(a_1,b_1)\times U(a_2,b_2)\times \ldots U(a_n,b_n)$ outputs realizations in a hypercube, it may be needed to ensure that the summation take only realizations within the hyperball $B(x,\delta x)$. Here $a_i = x_i - \delta x_i$ and $b_i = x_i + \delta x_i$. Presently, since $\mathfrak{g}_{ij} = \frac{\partial f}{\partial x_i}\frac{\partial f}{\partial x_j}$, we obtain the derivative $\frac{\partial f}{\partial x_i}$ from the following average:

$$\frac{\partial f(x)}{\partial x_i} = \frac{1}{N}\sum_j f(y_j)\frac{\partial \psi(|y_j - x|)}{\partial x_i}, i = 1, 2, \ldots \tag{5.36}$$

thereby relieving the original function of the need to be differentiated. To obtain the matrices associated with the Christoffel symbols, one needs $\frac{\partial \mathfrak{g}_{ij}}{\partial x_k}$ which is:

$$\frac{\partial \mathfrak{g}_{ij}}{\partial x_k} = \frac{\partial f}{\partial x_i}\frac{\partial^2 f}{\partial x_k \partial x_j} + \frac{\partial^2 f}{\partial x_k \partial x_i}\frac{\partial f}{\partial x_j} \tag{5.37}$$

where the second-order derivative may be obtained as:

$$\frac{\partial^2 f(x)}{\partial x_k \partial x_j} = \frac{1}{N}\sum_j f(y_j)\frac{\partial^2 \psi(|y_j - x|)}{\partial x_k \partial x_j}, i = 1,2,\ldots \tag{5.38}$$

The evaluations of the two derivatives in Equations (5.36) and (5.38) require the respective derivatives of the radial basis function $\psi(.)$. The latter exercise poses little difficulty due to these functions being sufficiently smooth. The following example illustrates the application of this approach in computing the $\mathfrak{g}$ matrix and its derivatives.

Example 5.3. We reconsider minimization of the 40-dimensional Rastrigin function in Example 5.2 and as given in Equation (5.32).

Solution. The bump function in Equation (5.33b) is taken as the RBF in approximating the $\mathfrak{g}$ matrix and its derivatives. The first- and second-order derivatives of the bump function required in Equations (5.36) and (5.38) are given in Appendix 5. With $\mathfrak{g}$ and the connection matrices involving the Christoffel symbols so obtained using the strategy enumerated above, results by GALA are shown in Figure (5.9) with $N = 10$. It is noticeable that the results compare well with those obtained (see Figure 5.8) using closed form expressions for $\mathfrak{g}$ and the connection matrices.

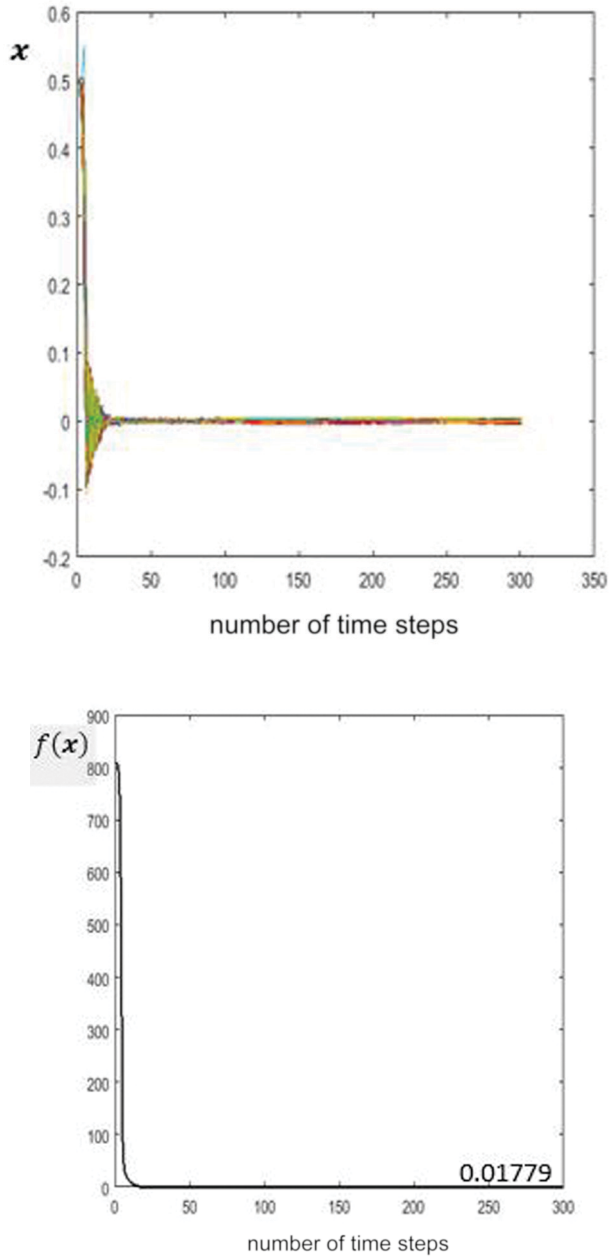


FIGURE 5.9 Optimization by GALA of a 40 -dimensional Rastrigin function with stochastic development and '$\mathfrak{g}$' matrix and its derivatives numerically computed by use of RBFs; (a) evolution of the solution x of the stochastically developed SDE (5.43) and (b) evolution of the objective function $f(x)$, $dt = 0.01$, $N_p = 5$.

■

5.4 PARAMETER ESTIMATION BY GALA

We present in this section an application of GALA to a parameter estimation problem which primarily consists in estimating the parameters of a given probability distribution using observed data. The topic has already been introduced in Chapter 3 (Section 3.2.2) and Chapter 4 (Section 4.4). Specifically, in Chapter 4, we have posed it as an optimization problem in a Riemannian setting. The statistical model comprising of the *pdf*s and parameterized by the unknown variables in the given distribution is treated as a manifold structure. It is a Riemannian manifold with a metric given by the KL divergence (Equation 4.97) which is shown to be equivalent to the FIM – an inner product on the manifold. Now, in the context of classical MALA (Section 4.5.1, Chapter 4), the evolution of the parameter vector $\theta(t)$ is governed by the Langevin SDE:

$$d\theta_t = \nabla l\left(\theta_t; Z\right)dt + dB_t \tag{5.39a}$$

$l\left(\theta_t; Z\right)$ is the log-likelihood function in Equation (4.93) which is rewritten below:

$$l(\theta_t; \boldsymbol{Z}) = \sum_{i=1}^{n} \log \mathbb{f}_{Z_i}(z_i; \theta_t) \tag{5.39b}$$

$\nabla = \dfrac{\partial}{\partial \theta}$ is the Euclidean gradient. Z is the vector of RVs $Z_i, i = 1, 2, \ldots, n$ corresponding to which the available observed data set is $z = \left\{z_1, z_2, \ldots, z_n\right\}^T$. Note that $\theta_t := \theta(t) \in \mathbb{R}^m$ comprises the unknown parameters in the *pdf* $\mathbb{f}_{\boldsymbol{Z}}(.)$ and need to be estimated. If one uses a Riemannian gradient in Equation (5.39) in lieu of the Euclidean one and further uses the exponential mapping at each time step along with a Metropolis test, it is RMALA, the Riemannian version of MALA (corresponding to Equation 4.111 in discretized form):

$$d\theta_t = \text{grad}\, l\left(\theta_t; Z\right)dt + dB_t \tag{5.40}$$

Now, according to GALA, we write the stochastically developed counterpart of the Langevin SDE in Equation (5.39a) as:

$$d\theta_t^i = \left[\sqrt{\mathfrak{g}^{-1}\left(\theta_t\right)}\right]_{ij} \left\{\nabla l\left(\theta_t; Z\right)\right\}^j dt - \frac{1}{2}\Gamma_{kj}^i\left(\theta_t\right)\left[\mathfrak{g}^{-1}\left(\theta_t\right)\right]_{kj} dt + \left[\sqrt{\mathfrak{g}^{-1}\left(\theta_t\right)}\right]_{ip} dB_t^p$$

$$i, j = 1, 2, \ldots, m \text{ and } p = 1, 2, \ldots, N \tag{5.41}$$

N is the dimension of the Brownian motion B_t. We call the Langevin SDE in the last equation as geometrically adapted, as the acronym GALA indicates. MALA,

making use of the Euclidean gradient in Equation (5.39a) to design a proposal distribution for the Markov chain may be considered as a first-order method. Loosely speaking, the scheme for GALA from Equation (5.41) is a 'second-order' one, as it makes use of derivatives up to the second order for the proposal step. Specifically, whilst working with the Langevin-diffusion based MCMC, a geometric adaptation of Langevin dynamics in GALA would enable us to restrict the evolving parameters on a hypersurface entirely consistent with the underlying constraints of motion. This in turn provides us with a handle to control the space-filling properties of Brownian motion that could be physically meaningless and often the cause of delayed convergence. Moreover, the modified drift term that restricts the solution of the Langevin equation to remain on the Riemannian hypersurface provides for an additional means of faster convergence and higher accuracy. The following two examples show the performance of GALA as applied to moderate and large dimensional estimation problems.

Example 5.4. We reconsider the parameter estimation problem in Example 4.8 of Chapter 4. The assumed *pdf* is the generalized exponential rewritten hereunder:

$$\mathbb{f}_Z(z;\theta) = \alpha\lambda e^{-\lambda z}(1-e^{-\lambda z})^{\alpha-1},\ z>0 \tag{5.42}$$

α and λ are the parameters to be estimated and hence the problem is two-dimensional with $m=2$ and $\theta=(\alpha,\lambda)^T$. Let $n=5000$ which is the number of observations $z_i, i=1,2,\ldots,n$ with reference (true) values for α and λ being 3.639 and 2.239, respectively.

Solution. The matrix $\mathfrak{g}$ corresponding to the Riemannian metric is given by the FIM $E_Z\left[\left.\frac{\partial^2 l(\theta;Z)}{\partial^2\theta}\right|_{\theta_k}\right]$ in Equation (4.102), i.e:

$$\mathfrak{g} = \begin{bmatrix} -\frac{n}{\alpha_k^{\,2}} & \sum_{i=1}^{n}\frac{z_i e^{-\lambda_k z_i}}{1-e^{-\lambda_k z_i}} \\ \sum_{i=1}^{n}\frac{z_i e^{-\lambda_k z_i}}{1-e^{-\lambda_k z_i}} & -\frac{n}{\lambda_k^2}-(\alpha_k-1)\sum_{i=1}^{n}\frac{z_i^2 e^{-\lambda_k z_i}}{\left(1-e^{-\lambda_k z_i}\right)^2} \end{bmatrix} \tag{5.43}$$

The Riemannian gradient $\operatorname{grad} l(\theta_t;Z)$ and the derivatives of $\mathfrak{g}$ with respect to the two parameters α and λ are given in Appendix 5. These are required in evaluating the Christoffel matrices $\Gamma, i=1,2$. Figure 5.10 shows the result by GALA of the estimated parameters α and λ. Result by RMALA (Section 4.6) is also included in the figure. Note that the iterations on the *x*-axis in the figure are pseudo time steps corresponding to the Langevin dynamics.

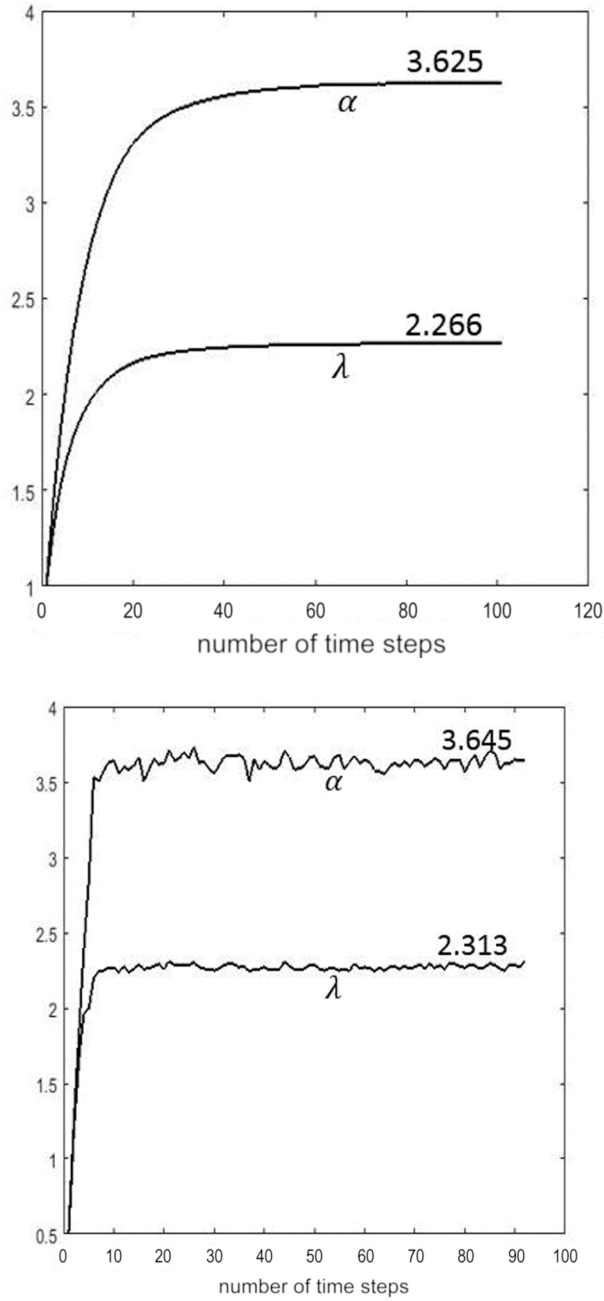


FIGURE 5.10 Statistical estimation by GALA of parameters of a generalized exponential probability distribution; evolution of parameters α and λ with iterations; (a) result by GALA and (b) result by RMALA (Section 4.7).

■

Example 5.5. We now consider a multivariate Gaussian distribution $N\left(\boldsymbol{\mu}, \boldsymbol{C}^{-1/2}\right)$ whose *pdf* is given by:

$$\mathbb{f}_{Z}(z;\theta) = \frac{\mathbf{1}}{(\mathbf{2\pi})^{m/2}\sqrt{|\boldsymbol{\Sigma}|}} \exp\left\{-\left(\frac{1}{2}(z-\mu)^{\boldsymbol{T}}\boldsymbol{\Sigma}^{-\mathbf{1}}(z-\mu)\right)\right\} \tag{5.44}$$

$\mu = \left(\mu_1, \mu_2, \ldots, \mu_m\right)^T \in \mathbb{R}^m$ s the mean vector of the joint RV $Z \in \mathbb{R}^m$. $|\Sigma|$ stands for the determinant of the covariance matrix $\Sigma \in \mathbb{R}^{m\times m}$ which is symmetric and positive definite. It is required to estimate the components of the unknown μ_Z and the matrix Σ by GALA given n observations of the multivariate normal distribution.

Solution. The number of parameters to be estimated is $D = m + m\left(m+1\right)/2$ with m mean values $\mu_1, \mu_2, \ldots, \mu_m$ and $\frac{m\left(m+1\right)}{2}$ covariance matrix components. Both these components constitute the vector $\theta \in \mathbb{R}^D$. As Σ is symmetric, the $\frac{m\left(m+1\right)}{2}$ components are the elements of either the upper or lower triangular matrix of Σ The estimation problem involves maximizing the log-likelihood function:

$$l(\theta; Z) = \sum_{i=1}^{n} \log \mathbb{f}_{Z_i}(z_i;\theta) \tag{5.45}$$

$z_i \in \mathbb{R}^m, i = 1, 2, \ldots, n$ is the i^{th} observation vector / data that follows the joint *pdf* $\mathbb{f}_{Z_i}(z_i;\theta)$ and is supposed to be available *a priori*. We refer to Mamajiwala and Roy [2022] for details on the Euclidean gradient $\frac{\partial l}{\partial \theta} = \left(\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, \ldots \frac{\partial l}{\partial \theta_m}\right)^T$ and the matrices associated with the Riemannian metric g and the Christoffel symbols.

Results for two- and three-dimensional problems are given in Figures 5.11 and 5.12 respectively along with the assumed reference (true) values. The results correspond to the estimates obtained by GALA for the D components of the mean vector and the covariance matrix. Note that $D = 5$ for the two-dimensional case $(m = 2)$ and $D = 9$ for the three-dimensional case $(m = 3)$.

Results by GALA for a large dimension of $m = 10$ are given in Figures 5.13 and 5.14. In this case, $D = 65$ with 10 mean components and 55 covariance matrix components (elements of upper or lower triangular matrix of $\boldsymbol{\Sigma}$). Only a few of these 65 components are shown in Figures 5.13 and 5.14. The reference (true) values are marked by dashed lines. Referring to Mamajiwala and Roy [2022], we find a comprehensive study on this parameter estimation problem of a normal *pdf* by GALA. Especially, the study therein illustrates the superior performance of GALA in estimating the parameters of large dimensional problems as against a number of other MCMC methods based on the manifold theory. The two metrics of performance used for comparison with other variants are efficiency and scalability.

■

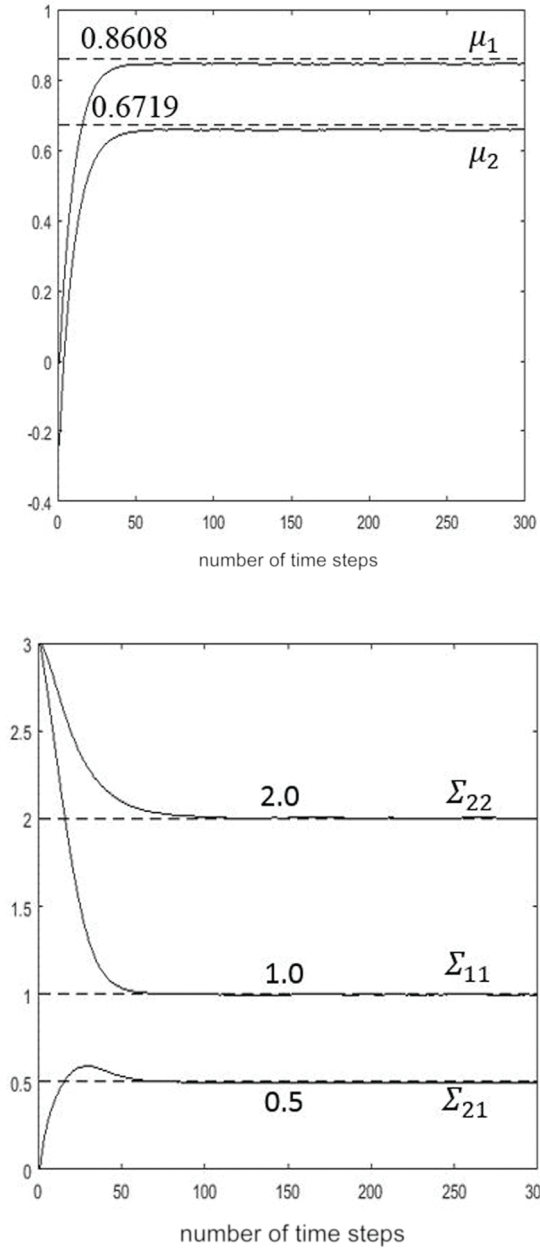


FIGURE 5.11 Estimation by GALA of parameters of a 2-dimensional normal pdf (Equation 5.44); evolution of (a) mean components μ_1 and μ_2 and (b) the covariance matrix components Σ_{11}, Σ_{21} and Σ_{22} with iterations; reference (true) values are shown by dashed lines.

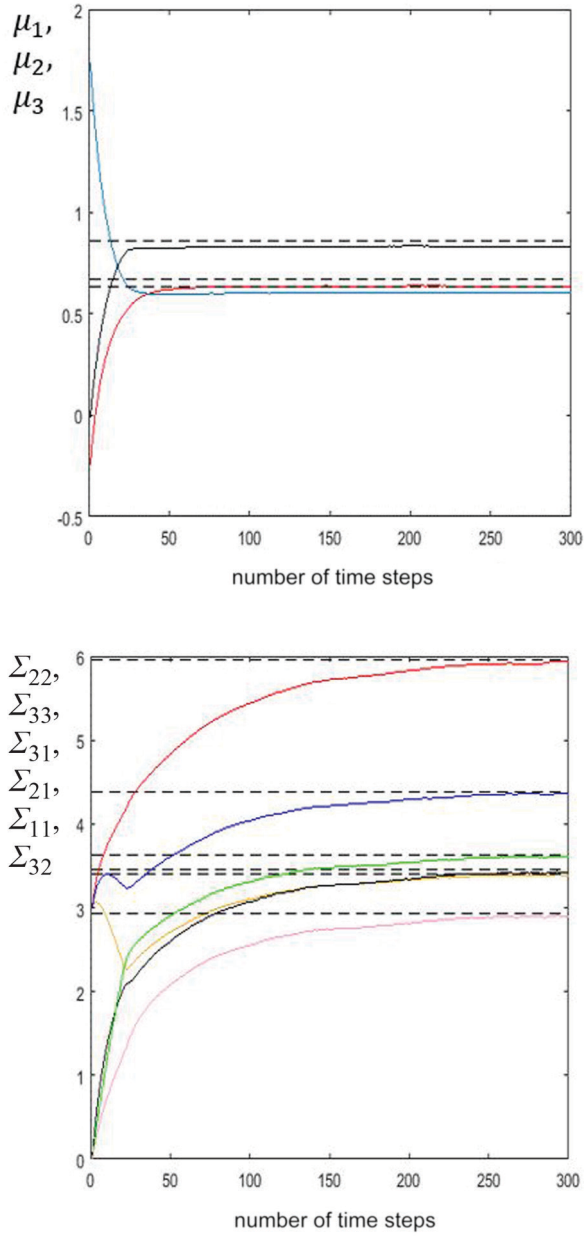


FIGURE 5.12 Estimation by GALA of parameters of a 3-dimensional normal pdf (Equation 5.44); evolution with iterations of a) mean components μ_1, μ_2 and μ_3 and (b) the covariance components Σ_{11} Σ_{21}, Σ_{22}, Σ_{31}, Σ_{32} and Σ_{33}; reference (true) values are shown by dashed lines – 0.8608 for μ_1, 0.6719 for μ_2 and 0.6309 for μ_3, 3.4084 for Σ_{21}, 3.488 for Σ_{21}, 5.9612 for Σ_{22}, 3.6288 for Σ_{31}, 2,9326 for Σ_{32}, 4.3737 for Σ_{33}

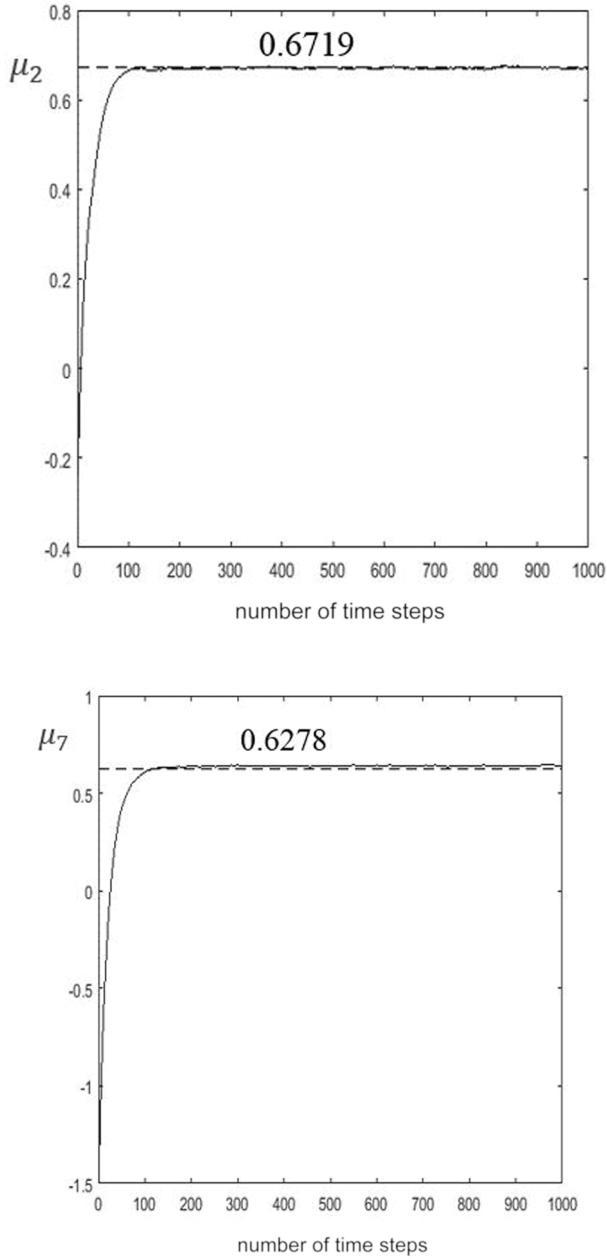


FIGURE 5.13 Estimation by GALA of parameters of a 10-dimensional normal pdf (Equation 5.44); evolution of (a) mean component μ_2 (reference value = 0.6719) and (b) mean component μ_2 with iterations; reference (true) values are shown by dashed lines.

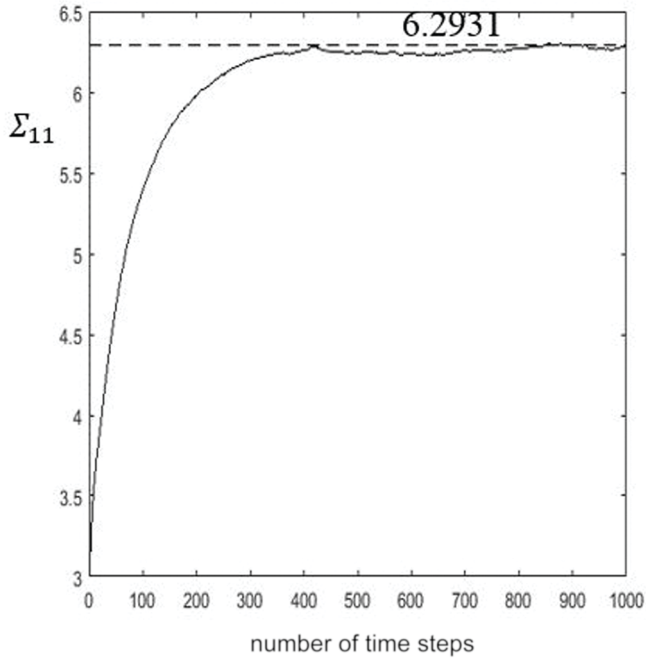
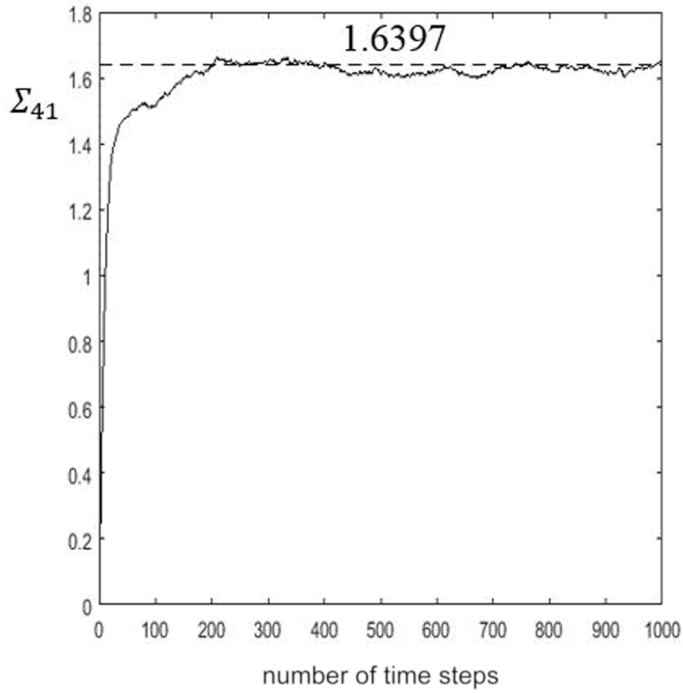


FIGURE 5.14a Estimation by GALA of parameters of a 10-dimensional normal *pdf* (Equation 5.44); evolution of (a) covariance matrix component Σ_{11} (reference value = 6.2931) and (b) covariance matrix component Σ_{41} (reference value = 1.6397); reference (true) values are shown by dashed lines. Estimation by GALA of parameters of a 10-dimensional normal *pdf* (Equation 5.44); evolution of (c) covariance matrix component Σ_{61} (reference value = –0.3291) and (d) covariance matrix component Σ_{74} (reference value = 0.1464); reference (true) values are shown by dashed lines. Estimation by GALA of parameters of a 10-dimensional normal *pdf* (Equation 5.44); evolution of (e) covariance matrix component Σ_{92} (reference value = 2.6237) and (f) covariance matrix component Σ_{96} (reference value = –1.7688); reference (true) values are shown by dashed lines.



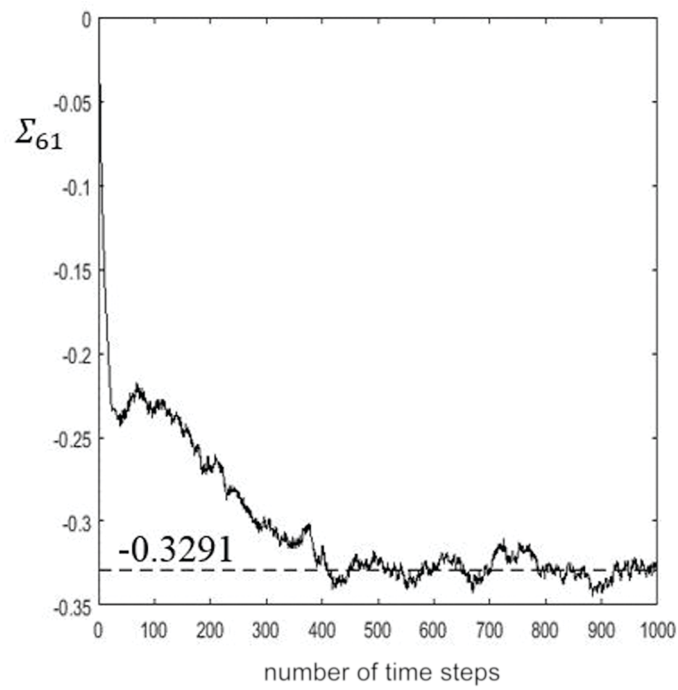
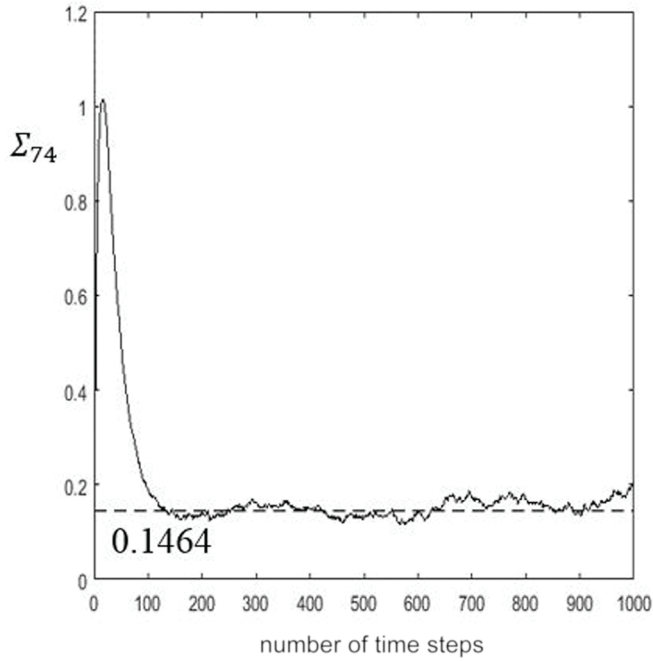


FIGURE 5.14b–c (Continued)



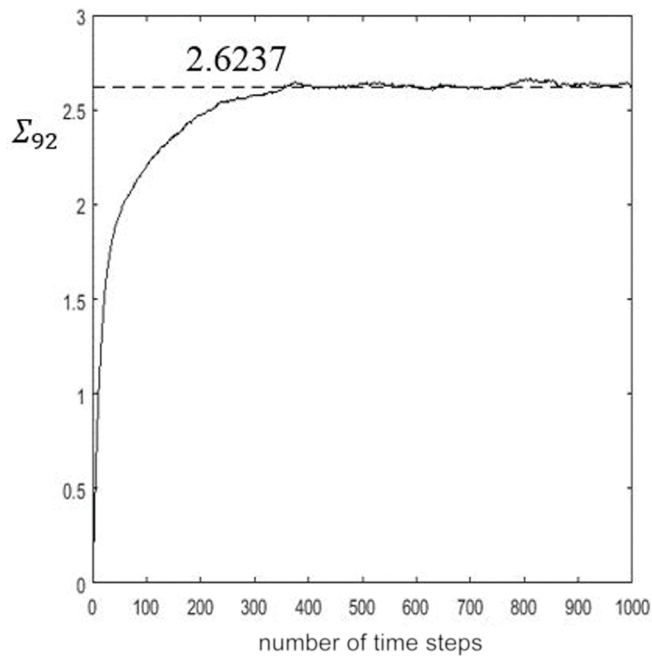


FIGURE 5.14d–e (Continued)

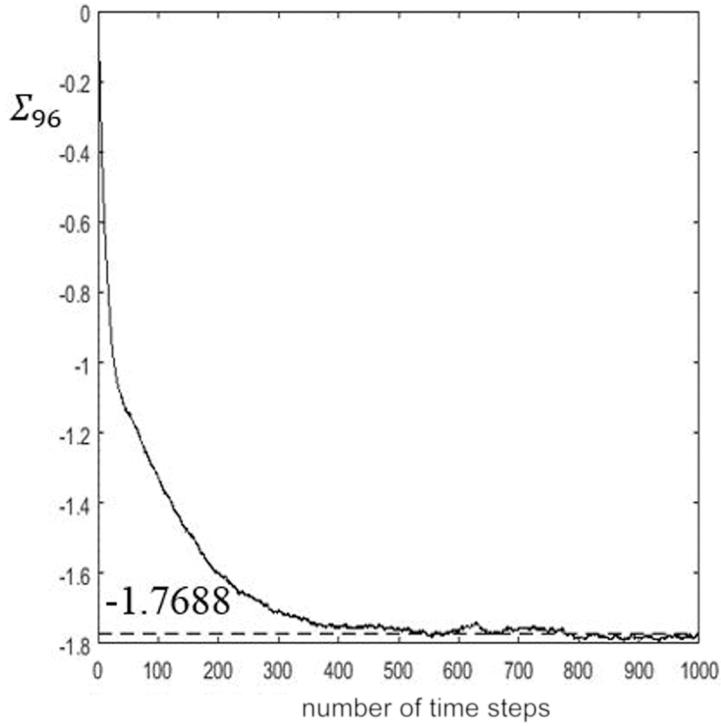


FIGURE 5.14f (Continued)

CONCLUDING REMARKS

As envisioned in the concluding part of the last chapter, we have proceeded in this chapter to geometrically adapt an SDE by making use of concepts from Riemannian differential geometry and stochastic calculus on manifolds. This adaptation known as the stochastic development and leading to the celebrated Laplace-Beltrami operator for a pure Brownian motion on a manifold has been highlighted. Underlying the stochastic development of a general SDE on a manifold lies the notion of an orthonormal frame bundle FM of a manifold M. A frame at a point on M is a linear isomorphism between the Euclidean space $R\mathbb{R}^d$ where the solution of a standard SDE evolves and the tangent space to M on which the solution needs to be projected. Thus, it is through the frame bundle that one can track the paths on the manifold once we know how it evolves in $\mathbb{R}^d$. Having systematically derived the stochastically developed SDE on a manifold, we have applied to the Langevin diffusion equation to arrive at a geometrically adapted Langevin dynamics and thus obtained the geometrically adapted version of MALA, which we have referred to as Geometrically Adapted Langevin Algorithm (GALA). The limitation or inconsistency on the part of RMALA described in the last chapter in representing a Riemannian version is removed in GALA. It is shown that the new version outperforms the classical MALA and RMALA in terms of faster convergence, higher accuracy and more importantly in its efficiency across a range of moderate and

large dimensional problems. Our specific illustrations have been with large dimensional optimization and parameter estimation problems.

EXERCISES

1. Show that for a vector field $X = X^i E_i \in T_p(M)$ expressed in terms of local coordinates on a manifold, the divergence $\text{div}\, X$ is given by:

$$\text{div}\, X = \frac{1}{\sqrt{|\mathfrak{g}|}} E_i \left(\sqrt{|\mathfrak{g}|}\, X^i \right) \text{ with } E_i = \frac{\partial}{\partial x^i},\ i = 1, 2, ..$$

2. Consider estimation by GALA, RMALA and MALA of parameter σ of a Rayleigh *pdf*:

$$f_Z(z;\sigma) = \frac{z}{\sigma^2} \exp\left(-\frac{z^2}{2\sigma^2}\right),\ \ 0 \le z \le \infty \tag{E5.1}$$

   where N samples or observations are available. Take the true parameter as $\sigma = 2$.
3. For a Weibull distribution with pdf given by:

$$f_Z(z;\ \lambda, k) = \frac{k}{\lambda}\left(\frac{z}{\lambda}\right)^{k-1} \exp\left(-\left(\frac{z}{\lambda}\right)^k\right),\ \ z \ge 0$$

$$= 0,\ z < 0 \tag{E5.2}$$

   estimate the two parameters λ and k by GALA, RMALA and MALA assuming the availability of N observations of the random variable z. True parameters for λ and k are respectively 1.0 and 1.5.
4. Solve by GALA, RMALA and MALA for the optimum of the n-dimensional Rosenbrock function:

$$f(x) = \sum_{j=1}^{n-1} 100\left(x_{j+1} - x_j^2\right)^2 + \left(1 - x_j\right)^2 \tag{E5.3}$$

5. Solve by GALA, RMALA and MALA for optimum x^* of the n-dimensional objective function (Himmelblau function):

$$f(x) = \frac{1}{n}\left(x_j^4 - 16x_j^2 + 5x_j\right) \tag{E5.4}$$

   (**Note**: Four optima x^* =[3.2, 2.0], [–2.805118, 3.131312], [–223 3.779310, –3.283186], and [3.584428, –1.848126] for the case with $n = 2$.

NOTATIONS

$B(t)$	Brownian motion
$e_1, e_2, \ldots$	canonical basis vectors of $\mathbb{R}^d$
$f(x)$	real valued smooth non-convex objective function
$\mathbb{f}_Z(z;\theta)$	*pdf*s in Equations (5.39b) and (5.44)
g	Riemannian metric
$H_e(y)$	horizontal field on *FM*
$H_y FM$	horizontal subspace on the frame bundle *FM*
$\mathfrak{T}_t$	Ito integral
L_t	infinitesimal generator
M	manifold
N_p	number of particles
q	a frame on FM
$Q=\left[Q_i^j\right]\in GL(d,\mathbb{R}),$	the general linear group
$T_p(M)$	tangent space on the manifold at a point p
$T_y(FM)$	tangent space at $y \in FM$, the frame bundle
$u:\mathbb{R}^d \to T_x(M)$	an isomorphism
U	a neighbourhood on a manifold
$\hat{U}=\pi^{-1}(U)$	a neighbourhood on a frame bundle
$V_y FM$	vertical subspace on the frame bundle *FM*
$x=\{x^i\}, i=1,2\ldots,d$	a local chart on M
$X_i=\dfrac{\partial}{\partial x^i}, i=1,2,\ldots,d$	coordinate basis vectors in $T_x M$
$X_{ij}=\dfrac{\partial}{\partial Q_j^i}, 1\le i,\ j\le d$	vectors spanning the vertical subspace $V_q FM$
$X(t)$	vector random process
$z_i\in\mathbb{R}^m, i=1,2,\ldots i^{th}$	observation vector / data in Examples 5.4
α	parameter to be estimated by MLE (Equation 5. 42)
$\alpha^i(\chi_t)$	drift term (Equation 5.16)
β_t	annealing parameter
$\mu\in\mathbb{R}^m$	mean vector (in the normal *pdf* – Equation 5.44)
λ	parameter to be estimated by MLE (Equation 5. 42)

π: $FM \to M$	a surjective map
$\sigma_j^i(\chi_t)$	diffusion term Equation 5.16)
γ_t	curve on the frame bundle *FM*
$\psi_t = \pi\gamma_t$	the projected curve on the manifold
$\psi(x)$	radial basis function (Equation 5.33a,b)
$\Sigma \in \mathbb{R}^{m\times m}$	covariance matrix (in the normal *pdf* – Equation 5.44)
$\theta(t)$	parameter vector (Equation 5.42)
χ_t	a curve in $\mathbb{R}^d$, an anti-development of ψ_t
∇	Euclidean gradient
Δ_E	Laplacian operator
Δ_M	Laplacian-Beltrami operator

REFERENCES

Belkin, M. and P. Niyogi. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems* 585–591.

Bromhead, D.S. and D. Lowe. 1988. Multivariable functional interpolation and adaptive networks. *Complex Systems* 2: 321–355.

Buhmann, M.D. 2003. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press. Cambridge.

Cohen, R. L. 1985. The immersion conjecture for differentiable manifolds. *Annals of Mathematics* 122(2): 237–328.

Das N. and M. C. Yip. 2020. Forward kinematics kernel for improved proxy collision checking. *IEEE Robotics and Automation Letter* 5(2): 2349–2356.

Hsu, E. P. 2002. *Stochastic Analysis on Manifolds*. American Mathematical Society. USA.

Ito, K. 1951. On a formula concerning stochastic differentials. *Nagoya Mathematical Journal* 3: 55–65.

Kobayashi, S. and K. Nomizu. 1963. *Foundation of Differential Geometry*, Vol. 1. Inderscience Publishers. New York, London, Sydney.

Kühnel, L.2018. *Stochastic Modelling on Manifolds*. PhD thesis. University of Copenhagen.

Kühnel, L., S. Sommer, and A. Arnaudon. 2019. Differential geometry and stochastic dynamics with deep learning numerics. *Applied Mathematics and Computation* 356: 411–437.

Kühnel, L. and S. Sommer. 2017. Stochastic development regression on non-linear manifolds. In: *Information Processing in Medical Imaging: 25th International Conference*, pp. 53–64.

Levy, B. and H.(R). Zhang. 2010. *Spectral Mesh Processing*. SIGGRAPH Asia Course.

Liu, D., G. Xu, and Q. Zhang. 2008. A discrete scheme of Laplace-Beltrami operator and its convergence over quadrilateral meshes. *Computers and Mathematics with Applications* 55(6): 1081–1093.

Mamajiwala M, and D. Roy. 2022. Stochastic dynamical systems developed on Riemannian manifolds. *Probabilistic Engineering Mechanics* 67(5)

Mamajiwala M, D. Roy, and S. Guillas. 2022. Geometrically adapted Langevin dynamics for Markov chain Monte Carlo simulations. arXiv:2201.08072v1

Manton, J. H. 2013. A primer on stochastic differential geometry for signal processing. *IEEE Journal of Selected Topics in Signal Processing* 7(4): 681–699.

Moody, J. and C. J. Darken. 1989. Fast learning in networks of locally-tuned processing units. *Neural Computation* 1: 281–294.

Moon, W. and Wettlaufer J. S. 2014. On the interpretation of Stratonovich calculus. *New Journal of Physics* 16: 1–13.

Powell, M. J. D. 1981. *Approximation Theory and Methods*. Cambridge University Press.

Roy D. and G. V. Rao. 2017. *Stochastic Dynamics, Filtering and Optimization*. Cambridge University Press.

Rustamov, R.M. 2007. Laplace-Beltrami eigenfunctions for deformation invariant shape representation. *Proceedings of the Fifth Eurographics Symposium on Geometry Processing, Eurographics Association,* pp. 225–233.

Shapiai, M. I., Z. Ibrahim, M. Khalid, L. W. Jau, V. Pavlovic, and J. Watada. 2011. Function and surface approximation based on enhanced kernel regression for small sample sets. *International Journal of Innovative Computing, Information and Control* 7(10): 5947–5960.

Shawe-Taylor J. and N. Christianini. 2004. *Kernel Methods for Pattern Analysis.* Cambridge: Cambridge University Press.

Sommer, S. and A. M. Svane. 2017. Modelling anisotropic covariance using stochastic development and sub-Riemannian frame bundle geometry, *American Institute of Mathematical Sciences* 391–410.

Stratonovich, R. L. 1966. A new representation for stochastic integrals and equations. *SIAM Journal on Control and Optimization* 4: 362–371.

Tang, K., X. Li, P. N. Suganthan, Z. Yang, T. Weise. 2009. Benchmark functions for the CEC'2010 special session and competition on large scale global optimization. *Nature Inspired Comput. Applicat. Lab., Tech. Rep.*

Urakawa, H. 1993. Geometry of Laplace-Beltrami operator on a complete Riemannian manifold. *Advanced Studies in Pure Mathematics. Progress in Differential Geometry* 22: 347–406.

van Kampen, N. G. 1981. Itô versus Stratonovich. *Journal of Statistical Physics* 24: 175–87.

Yi, S., Hamid-Krim, and L. K. Norris. 2011. Human activity modeling as Brownian motion on shape manifold. *International Conference on Scale Space and Variational Methods in Computer Vision* 628–639.

BIBLIOGRAPHY

Andrieu, A., N. De Freitas, A. Doucet, and M. I. Jordan. 2003. An introduction to MCMC for machine learning. *Machine Learning* 50(1–2): 5–43.

Xifara, T., C. Sherlock, S. Livingstone, S. Byrne, and M. Girolami. 2014. Langevin diffusions and the Metropolis-adjusted Langevin algorithm. *Statistics and Probability Letters* 91(C): 14–19.

Appendix 1

A1.1 COMPUTATIONAL COMPLEXITY AND NP HARD OPTIMIZATION PROBLEMS

The class of problems, P, NP, NP hard and NP complete, signify the complexity of computation (see Figure A1.1). A formal definition of these classes can be found in (Garey and Johnson 1979, Leeuwen 1998). The class P corresponds to those problems that are solvable in polynomial time. Suppose that an algorithm runs in $O(n)$ time where n is problem size, the run time is said to be a linear function of n. Similarly, one can have an algorithm that runs in quadratic time $O(n^2)$, cubic time $O\left(n^3\right)$ and, in general, in polynomial time and we then say that the problems it solves are said to be in class P.

There exist algorithms that do not run in polynomial time on regular computers, but run in polynomial time on a non-deterministic Turing machine (Martin 1997). These programs belong to NP that stands for non-deterministic polynomial time. Equivalently, NP defines problems (decision problems) that can be verified in polynomial time. This does not necessarily mean that there is a polynomial-time way to find a solution. Intuitively NP is a class of decision problems where one can verify an answer quickly in reasonable time if one has been provided with a solution. For example, consider the Hamiltonian path problem. The Hamiltonian path is a path – in an undirected or directed graph $G\left(V,E\right)$ where V denotes the vertices and E the edges – that visits each vertex exactly once. Assume that we have a solution path on hand. Verifying if it is correct can be performed in a polynomial time. Note that all problems in P are also in NP.

A problem $\mathcal{A}$ is NP-complete, if

(i) it is in NP, i.e. if it is polynomial-time verifiable and
(ii) every problem in NP is reducible to $\mathcal{A}$ in polynomial time.

As an example, consider a travelling salesman problem (TSP) – the minimum Hamiltonian cycle of cost $\leq k$, where each edge has an associated weight. A Hamiltonian cycle is a simple cycle in a graph that visits all vertices exactly once before returning to the start vertex. One can verify in polynomial time if a cycle

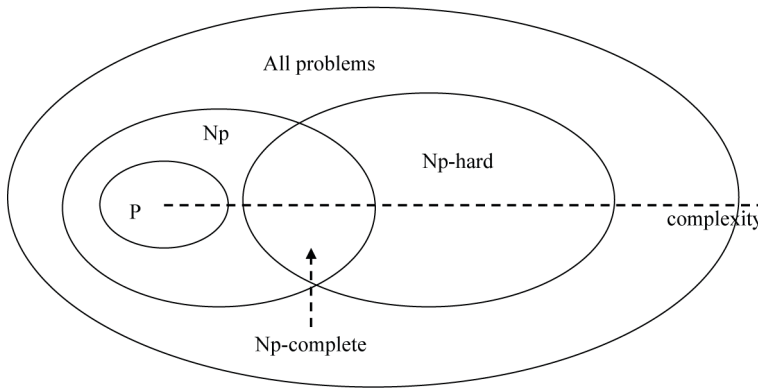


FIGURE A1.1 Categorization of optimization problems – P, NP, NP-complete and NP-hard.

visits all vertices and has cost $\leq k$ and hence a TSP is NP. It is NP-complete since a known NP-complete problem such as a Hamiltonian cycle can be reduced to the TSP (Garey and Johnson 1979). It is not known if every problem in NP can be solved and this is called the P versus NP problem (Lance 2009). However, if any NP-complete problem can be solved, then every problem in NP can be solved because of condition (ii) appearing in the definition of NP-completeness.

A problem satisfying condition (ii) above is an NP-hard problem whether it satisfies the condition stated in (i) or otherwise. NP-hard problems are the hardest of all problems in NP. They belong to a class where, even when one has a solution, it cannot be verified in polynomial time. More precisely, a problem is called NP-hard if it is polynomial-time reducible but not necessarily polynomial-time verifiable. TSP where one needs to find the shortest distance (cycle) covered, is an example of an NP-hard problem. That is, given a weighted graph *G*, the task is to find the shortest cycle (an optimization problem) that visits every vertex. Finding the shortest cycle is obviously harder than determining if a cycle exists at all; so the TSP is NP-hard.

A1.2 METRIC *d*(*x*, *y*) AND ITS PROPERTIES

Given a metric space (X,d) with X being a set associated with a metric $d(x,y)$ where $x,y \in X$. The metric $d(x,y)$ is a function from $X \times X$ to R such that the following conditions hold for every $x,y,z \in X$

(i) Non-negativity: $d(x,y) \geq 0$

(ii) Symmetry: $d(x,y) = d(y,x)$

(iii) Triangle inequality: $d(x,y) + d(y,z) \geq d(x,z)$

(iv) $d(x,y) = 0$ if and only if $x = y$

Euclidean space is a familiar example for a metric space. Space $\mathbb{R}^n$ is equipped with the Euclidean distance $d(x,y) = \|x-y\|_2 = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$. All inner product and normed spaces are metric spaces.

A1.3 BASIC PROBABILITY THEORY AND RANDOM NUMBER GENERATION

In the search for a solution to TSP by Metropolis algorithm (Section 1.3.3, Chapter 1), some basic concepts of probability are utilized. To familiarize readers with the bare bones of probability theory (Papoulis 1991), i.e. random variables and the associated probability distributions, we briefly recapitulate here these aspects (Papoulis 1991) and also describe a scheme for random number generation. The random number generation is a primary requirement in a simulation study which is the mainstay of stochastic optimization – a subject of this book. In Appendix 4, we briefly describe the theory of stochastic processes which closely relates to the concepts in probability outlined here and forms the basis for geometrical methods of optimization described in Chapters 4 and 5.

A1.3.1 Random Variables and Probability Distributions

Any outcome of an observation or experiment (viz. of number of vehicles crossing a busy traffic junction or wind velocity measurement at a location every one hour) may often have a random character in that a sequence of such outcomes may not follow a deterministic or predictable pattern. Instances one may be familiar with are the outcomes of a coin tossing experiment, throw of a dice and rainfall on a day in the rainy season. One may describe the possible outcomes on rainfall observation on a day as high, medium or low. The outcomes from a coin tossing experiment could be 'head' or 'tail' and, from a dice throw, any of the six faces $(f_1, f_2, f_3, f_4, f_5, f_6)$. These outcomes contribute to form the sample space Ω. The complement of (the set of) all non-trivial outcomes in Ω is the null outcome ϕ. In these simple cases of random experiments (viz. where outcomes are discrete and their number is finite), it is possible to assign a probability to each of the outcomes. For example, in an unbiased coin tossing experiment, probability P of 'head' or 'tail' may be ½ each so that $P(\Omega) = 1$. Similar may be the case with the throw of a fair dice with $P(f_1) = P(f_2) = P(f_3) = P(f_4) = P(f_5) = P(f_6) = 1/6$. The probability of the null event ϕ is set to zero. To facilitate a more rational description and easy mathematical manipulation of these random quantities in a unified fashion, the concept of a random variable is introduced. Using a random variable, the outcomes in Ω which may be different from real numbers (or, perhaps, vectors of real numbers) are made to correspond to subsets of the real line. If a random variable is denoted by X, it thus denotes a mapping of Ω to $\mathbb{R}$, i.e., $X: \Omega \to \mathbb{R}$. For example, the two outcomes 'head' and 'tail' in the coin tossing experiment may respectively be mapped to, say, 0 and 1 on

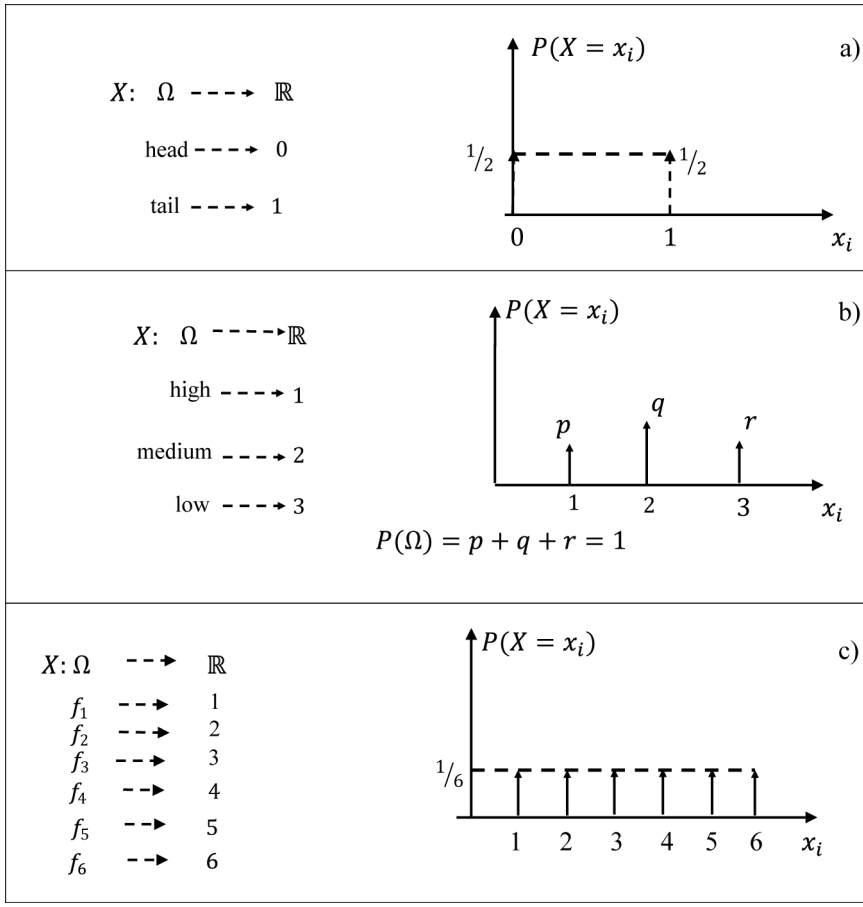


FIGURE A1.2 Random variables and probabilities: (a) tossing of an unbiased coin; (b) rainfall on a day and (c) throw of an unbiased dice.

the real line. The three outcomes 'high', 'medium' and 'low' of rainfall on a day may be mapped to 1, 2 and 3, respectively. If the assigned probabilities for the three possibilities of rainfall are $P(1) = p$, $P(2) = q$ and $P(3) = r$, one must have $p + q + r = 1$. Figure A1.2 pictorially shows these random variables along with their probabilities.

In a general probability setting, a numerical or computational treatment of uncertainty requires that the outcomes or events be made to correspond one-to-one with real numbers via an appropriate mapping from Ω to $\mathbb{R}^n$. A random variable $\boldsymbol{X} \in \mathbb{R}^n$ describes this mapping and is usually associated with a triplet $(\Omega, \mathcal{F}, \mathrm{P})$ known as the probability space. $\mathcal{F}$ is known as the sigma algebra (written as σ-algebra). It is defined as a non-empty set consisting of all observable subsets (events) belonging to Ω closed under complementation and finite unions of its subsets including Ω and the null outcome φ. For example, in a coin tossing experiment with $\Omega = \{s, f\}$

where s and f stand for a success and failure, $\mathcal{F}$ is the power set containing all possible events $\{\varphi, s, f, \Omega\}$. Similarly, for the dice throw example, $\mathcal{F}$ is the set of all $= 2^6 = 64$ events possible out of the sample space $\Omega = \{f_1, f_2, f_3, f_4, f_5, f_6\}$.

A1.3.2 Discrete Random Variables

The three examples in Figure A1.2 represent discrete random variables as the outcomes are countable. When each outcome in Ω is considered as being equally likely, the random variable is said to follow a uniform probability distribution. The dice throw in Figure A1.2c is an example of such a uniformly distributed random variable defined in a discrete set-up. In a TSP with N cities, consider picking a city at random. With $\Omega = \{city\ 1, city\ 2, \ldots, city\ N\}$, let $X{:}\Omega \to \{1, 2, \ldots, N\}$. If we assign $\mathrm{P}(X = 1\ or\ 2 \ldots \text{or}\ N) = \frac{1}{N}$, then X is uniformly distributed. The plots of $P(X = x)$ in Figure A1.2 are actually those of probability density (or mass) functions (*pdf*s) of the three random variables shown therein. Henceforth, a *pdf* will be denoted by $\mathbb{f}_X(x)$ where the argument x stands for a realization of the random variable X (note that x belongs to the range of X). We define $\mathbb{f}_X(x)$ for a discrete random variable as:

$$\mathbb{f}_X(x) = \sum_j P(X = x_j)\delta(x - x_j) \tag{A1.1}$$

$\delta(.)$ stands for a Dirac delta function. The Dirac delta function is a 'generalized function' in that $\int_{-\infty}^{\infty} g(x)\delta(x-a)dx = g(a)$. A heuristic definition is:

$$\begin{aligned} \delta(x-a) &= \infty, \text{ for } \quad x = a \\ &= 0, \qquad \text{otherwise} \end{aligned} \tag{A1.2a}$$

with the constraint

$$\int_{\mathbb{R}} \delta(x-a)dx = 1 \tag{A1.2b}$$

The cumulative probability distribution function (CDF) of X, denoted by $\mathbb{F}_X(x)$, is defined for a discrete random variable, as:

$$\mathbb{F}_X(x) = P(X \le x) = \sum_{j:x_j \le x} P(X = x_j) \tag{A1.3a}$$

An alternative way of expressing $\mathbb{F}_X(x)$ is:

$$\mathbb{F}_X(x) = \sum_j P(x_j)\mathbb{U}(x - x_j) \tag{A1.3b}$$

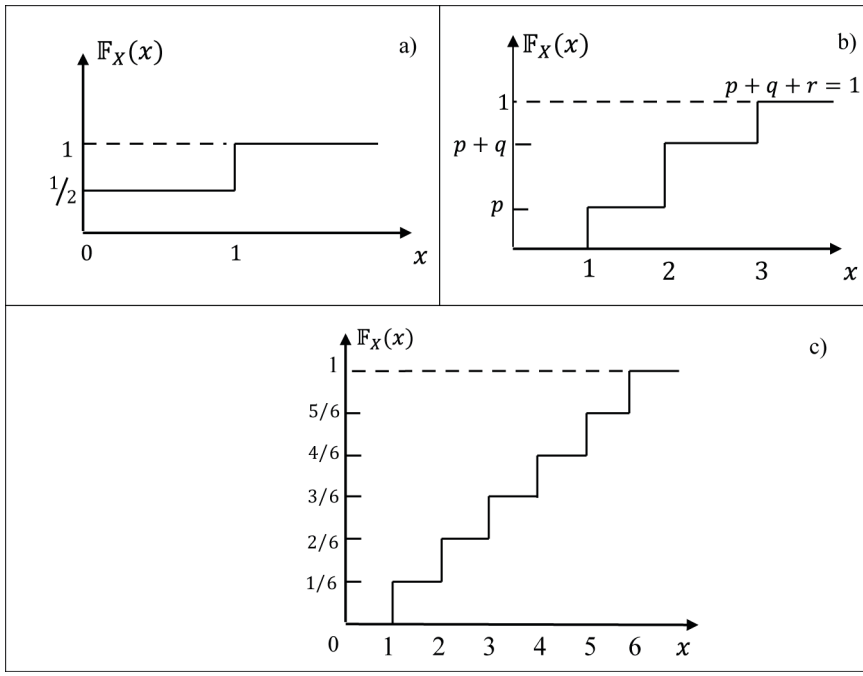


FIGURE A1.3 Discrete random variables and CDFs: (a) unbiased coin tossing; (b) rainfall on a day and (c) throw of an unbiased dice.

where $\mathbb{U}\left(x - x_j\right)$ is the unit step function defined by:

$$\mathbb{U}\left(x - x_j\right) = 1, \quad \text{for } x \geq x_j$$

$$= 0, \quad \text{for } x < x_j \tag{A1.4}$$

$\mathbb{F}_X\left(x\right)$ is thus the cumulative total (sum or integral as the case may be) of probabilities up to x. From Equation (A1.3), $\mathbb{F}_X\left(-\infty\right) = P\left(\Phi\right) = 0$ and $\mathbb{F}_X\left(\infty\right) = P\left(X \leq \infty\right) = P\left(\Omega\right) = 1$. For the three random variables in Figure A1.2, the CDFs are shown in Figure A1.3.

A1.3.3 Continuous Random Variables

Consider the case of a Roulette wheel, spinning with a pointer that can stall at any angle Θ within $0 - 2\pi$ radians. Clearly, Θ could be any real number in $[0, 2\pi]$. The outcomes are uncountably infinite and Θ is called a continuous random variable. If

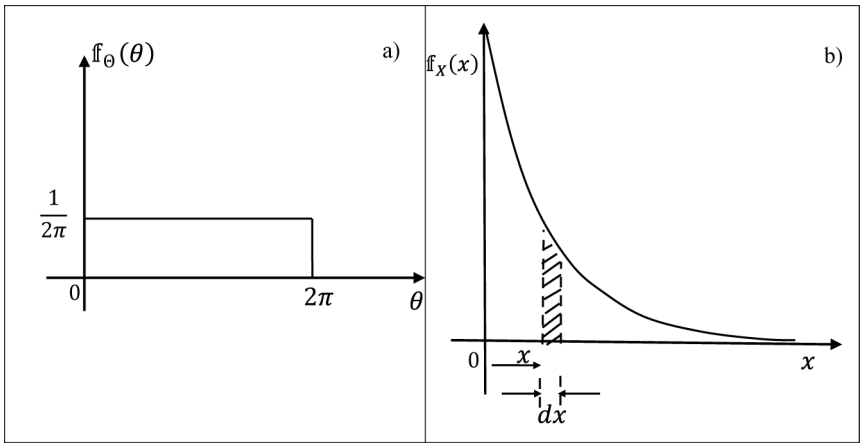


FIGURE A1.4 (a) *pdf* of a uniformly distributed (continuous) random variable $\Theta \in [0, 2\pi]$ – Roulette wheel experiment and (b) *pdf* of an exponential (continuous) random variable with $\mathbb{f}_X(x) = e^{-x}$.

the outcomes are assumed to be equally likely, Θ is said to follow a uniform probability distribution over $[0, 2\pi]$ (see Figure A1.4a).

In general, for a uniformly distributed continuous random variable X, $\mathbb{f}_X(x)$ is defined by:

$$\mathbb{f}_X(x) = \frac{1}{b-a}, \quad a \le x \le b$$

$$= 0, \quad \text{otherwise} \tag{A1.5a}$$

A uniformly distributed random variable in the interval $[a,b]$ is usually denoted by $U(a,b)$.

Figure A1.4b shows an exponential random variable with *pdf:*

$$\mathbb{f}_X(x) = e^{-x}, \quad x > 0 \tag{A1.5b}$$

In fact, the Boltzmann distribution in Equation (1.10) in Chapter 1 is an exponential distribution which is a discrete analogue of the *pdf* in Equation (A1.5b). From Figure A1.4(b), we observe that $P(x \le X \le x + dx)$ is given by the hatched area:

$$P(x \le X \le x + dx) = \mathbb{f}_X(x)dx \tag{A1.6}$$

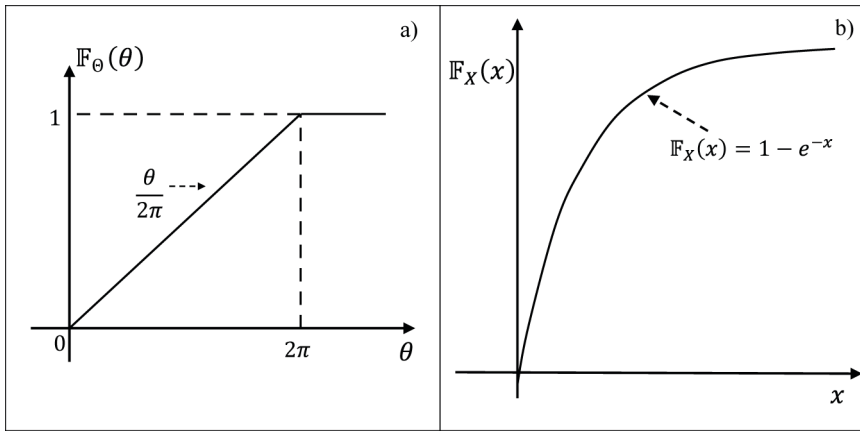


FIGURE A1.5 CDFs of the continuous random variables: (a) uniformly distributed and (b) exponentially distributed (see the corresponding *pdf*s in Figure A1.4).

For a continuous random variable, the CDF is defined as:

$$\mathbb{F}_X(x) = P(X \leq x) = \int_{-\infty}^{x} \mathbb{f}_X(x)dx \tag{A1.7}$$

Thus, in defining $\mathbb{F}_X(x)$ for a continuous random variable, an integral replaces the summation in Equation (A1.3) of the discrete case. The properties of the CDF follow:

$$\mathbb{F}_X(-\infty) = P(X \leq -\infty) = 0 \tag{A1.8a}$$

and

$$\mathbb{F}_X(\infty) = P(X \leq \infty) = \int_{-\infty}^{\infty} \mathbb{f}_X(x)dx = \text{area under the } pdf \text{ curve} = 1 \tag{A1.8b}$$

The inference from Equation (A1.7) is that $\mathbb{f}_X(x) = \frac{d\mathbb{F}_X(x)}{dx}$, provided the derivative exists. CDFs for the uniformly distributed and exponential random variables are shown in Figure A1.5.

Another familiar distribution is normal or Gaussian distribution with *pdf* given by:

$$\mathbb{f}_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{A1.9}$$

μ and σ are parameters in the distribution. Normal random variables are commonly encountered in engineering applications. For instance, measurement errors in experiments (or 'noise' as it is commonly called) are often modelled by normal (Gaussian) distributions. Moreover, the normal distribution also derives its importance from the central limit theorem (CLT) which implies that the sum of a large number of independent and identically distributed random variables is approximately normal. We refer to Roy and Rao (2017) for more details on the CLT.

For a discrete random variable, non-zero point probabilities exist, as is evident from Figure A1.2. But in the case of continuous random variables, a point probability is zero (Roy and Rao 2017). This is analogous to the common knowledge that the mass at a point in a body is zero. For a continuous random variable, one defines the probability of X within an interval as in Equation (A1.6).

A1.3.4 Expectation of Random Variables

If X is a random variable defined on a sample space Ω with a probability measure P, then the expectation of X denoted by $E_P[X]$ is defined by the following (Lebesgue) integral with respect to P:

$$E_P[X] = \int_\Omega X dP = \int_\Omega X(\omega) dP(\omega) \tag{A1.10}$$

If X has the probability density function $\mathbb{f}_X(x)$, then we have:

$$\int_\Omega X dP = \int_{\mathbb{R}} x dF_X(x) = \int_{\mathbb{R}} x \mathbb{f}_X(x) dx \tag{A1.11}$$

$E_P[X]$ is the first (order) moment as given by Equation (A1.10) and is known as the expectation or mean of *X*. The subscript '*P*' in $E_P[.]$ is usually omitted unless there is a scope for confusion with different probability measures. The expectation converts a random variable to a 'weighted average' in that if one performs the random experiment of sampling X indefinitely many times, then the average of the resulting numbers approaches $E[X]$. The above definition applies to both discrete and continuous random variables except that, for discrete *X*, summation replaces integration as:

$$E[X] = \sum_k x_k f_x(x_k) \tag{A1.12}$$

Extending the definition of expectation of a random variable X to higher order moments, one has an n^{th} order moment $E[X^n] = \int_{-\infty}^{\infty} x^n f_x(x) dx$ in the case of a continuous random variable and $E\left[(X-\mu)^m\right] = \sum_{k=1}^{n} x_k{}^m f_x(x_k)$ in the discrete case. Central moments are given by $\int_{\mathbb{R}} (x-\mu)^n \mathbb{f}_X(x) dx$. It follows that the variance of X is defined by:

$$\sigma^2 = \int_{\mathbb{R}} (x-\mu)^2 \mathbb{f}_X(x) dx \tag{A1.13}$$

σ is the standard deviation. For example, the two parameters μ and σ in the *pdf* in Equation (A1.9) represent respectively the first order moment (the mean) and the second order moment (the standard deviation) of a normal random variable.

$\mathcal{N}(\mu, \sigma)$ is the notation generally used to indicate a normal probability distribution. A generalization of the definition of expectation is to include a (deterministic) function $g(X)$ of a random variable in that one may have:

$$E\left[g(X)\right] = \int_{\mathbb{R}} g(x) f_X(x) dx \tag{A1.14}$$

A1.3.5 Independence of Random Variables

The concept of independence is fundamental in probability theory. Two events E_1 and E_2 are said to be independent if the probability of both events occurring together – expressed by $P(E_1 \cap E_2)$ – is given by:

$$P(E_1 \cap E_2) = P(E_1) P(E_2) \tag{A1.15}$$

Instead, if occurrence of event E_1 is dependent on E_2, the dependence may be expressed in terms of a conditional probability denoted by $P(E_1 | E_2)$ as:

$$P(E_1 \mid E_2) = \frac{P(E_1 \cap E_2)}{P(E_2)}$$

$$\Rightarrow P(E_1 \cap E_2) = P(E_1 \mid E_2) P(E_2) \tag{A1.16a}$$

Clearly, in view of Equation (A1.15), independent events E_1 and E_2 are characterized by:

$$P(E_1 \mid E_2) = P(E_1) \text{ and } P(E_2 \mid E_1) = P(E_2) \tag{A1.16b}$$

For example, knowing that the event $E =$ an even number appeared in a dice throw, the probability $P(2|E) = P(4|E) = P(6|E) = \frac{1}{3}$ and $P(1|E) = P(3|E) = P(5|E) = 0$.

Now given two random variables (RVs) X and Y, we say that they are independent if the events $X \le x$ and $Y \le y$ are independent, i.e. if:

$$P(X \le x \cap Y \le y) = P(X \le x) P(Y \le y) \tag{A1.17}$$

The LHS of the last equation is usually denoted by $\mathbb{F}_{XY}(x, y)$ which is known as the joint probability (cumulative) distribution function. So X and Y are independent if:

$$\mathbb{F}_{XY}(x,y) = \mathbb{F}_X(x)\mathbb{F}_Y(y) \tag{A1.18a}$$

A conditional probability distribution $\mathbb{F}_{X|Y}(x|y)$ is defined as:

$$\mathbb{F}_{X|Y}(x|y) = \frac{\mathbb{F}_{XY}(x,y)}{\mathbb{F}_Y(y)} \tag{A1.18b}$$

Equation (A1.18a) results in case X and Y are independent, i.e., if $\mathbb{F}_{X|Y}(x|y) = \mathbb{F}_X(x)$.

It follows that for two independent X and Y, the joint *pdf* of X and Y is given by:

$$\mathbb{f}_{XY}(x,y) = \mathbb{f}_X(x)\mathbb{f}_Y(y) \tag{A1.19a}$$

From the last equation, one may express the conditional *pdf* $\mathbb{f}_{X|Y}(x|y)$ as:

$$\mathbb{f}_{X|Y}(x|y) = \frac{\mathbb{f}_{XY}(x,y)}{\mathbb{f}_Y(y)} \tag{A1.19b}$$

A generalization follows for n independent RVs $X_1, X_2, \ldots, X_n$. The joint *pdf* is:

$$\mathbb{f}_{X_1X_2\ldots X_n}(x_1, x_2, \ldots, x_n) = \mathbb{f}_{X_1}(x_1)\mathbb{f}_{X_2}(x_2)\ldots\mathbb{f}_{X_n}(x_n) \tag{A1.20}$$

In case the RVs X and Y are not independent, they may be correlated. For instance, stock prices whose variations are stochastic show correlations as noticeable from observations over a period of time. The correlation between X and Y is expressed as $E[XY]$ where $E[.]$ is the expectation operator (Equations A1.10 and A1.11) and therefore:

$$E[XY] = \int_{\mathbb{R}}\int_{\mathbb{R}} xy\,\mathbb{f}_{XY}(x,y)dxdy \tag{A1.21}$$

It follows that correlated random variables are characterized by a covariance matrix:

$$\boldsymbol{C} = \begin{bmatrix} E\left[\left(X-\mu_X\right)^2\right] & E\left[\left(X-\mu_X\right)\left(Y-\mu_Y\right)\right] \\ E\left[\left(X-\mu_X\right)\left(Y-\mu_Y\right)\right] & E\left[\left(Y-\mu_Y\right)^2\right] \end{bmatrix} \tag{A1.22}$$

μ_X and μ_Y are the mean values of X and Y. Note that $\boldsymbol{C}$ is a symmetric matrix. While the diagonal terms are the individual variances σ_X^2 and σ_Y^2, the off-diagonal terms denote the covariance $\sigma_{XY} = cov(X,Y)$ between the random variables. Obviously, if the covariance term is zero, X and Y are uncorrelated. Independence implies uncorrelatedness while the converse may not be true (Papoulis 1991). The mean-variance portfolio theory of Markowitz (1952) is the basic model in finance which

uses correlation analysis for minimization of risk and to achieve better expected returns.

A1.3.6 Random Number Generation

A uniformly distributed random variable assumes importance in simulation studies – the example on TSP being a case in point. Today's computing machines are equipped with random number generators that provide random numbers uniformly distributed in $[0,1]$. In practice, it is difficult to generate truly random numbers of infinite sequence. The sequence of numbers generated on a computer only approximates the properties of a random variable. Also, the numbers started from a seed number are bound to have a periodicity, i.e. the same sequence of numbers repeat after a certain period. A common algorithm (Knuth 1997) used to generate the pseudo-random numbers is:

$$X_{j+1} = \left(AX_j + B\right) \bmod M \tag{A1.23}$$

Here A, B and M are non-negative integers. A is called the multiplier, B the increment and M the modulus. Equation (A1.23) corresponds to a linear congruential generator, i.e. $C \equiv D \bmod M$ which is read as 'C is congruent to D modulo M'. This amounts to $C = D - K \times M$ where K is the largest positive integer less than D/M, i.e. $K = \lfloor D/M \rfloor$ using the notation of the largest integer function. X_0, the starting element in the pseudo-random sequence, is known as the seed of the random number generator. The random numbers generated by Equation (A1.23) lie in $[0, M-1]$. At the end of n iterations, the recurrence formula in Equation (A1.23) gives:

$$X_n = \left(A^n X_0 + B \, \frac{A^{n-1} - 1}{A - 1} \right) \bmod M \tag{A1.24}$$

From the above, one obtains the period of the pseudo-number generator by setting $X_n = X_0$. Note that if the number of bits associated with a word size in a computer is m, the largest possible period is found to be $2^m = M$. From X_n generated through Equation (A1.24), we obtain the uniformly distributed pseudo-random number $\frac{X_n}{M} \in [0,1]$. Now, a realization for $U(a,b)$ can be obtained by the relation $U(a,b) = a + (b-a)\frac{X_n}{M}$. A random number generator is required to be computationally efficient and have a large period. For more details on random number generators and on tests to ensure a uniform distribution, we refer to L'Ecuyer (1992) and Nishimura (2000).

For solution to TSP by Metropolis algorithm, step (ii) in Table 1.4 requires generating a new trial solution $\hat{\boldsymbol{x}}_k$ at every iteration k.This is accomplished by the swapping of connections (Figure 1.12) of any two out of N cities. This requires picking two cities i and j randomly from the set $\{1,2,\ldots,N\}$. To perform this, we

independently generate two uniformly distributed numbers $U^{(1)}(0,1)$ and $U^{(2)}(0,1)$. Then $r_1 = NU^{(1)}(0,1)$ and $r_2 = NU^{(2)}(0,1)$ give real numbers in the interval $[0,N]$. i and j are now obtained, say, by rounding off r_1 and r_2 to the nearest integers greater than r_1 and r_2.

Here, a mention needs to be made about the significant role played by the uniform *pdf* in simulation studies. It is extensively used to generate random numbers with other (specified) distributions. This, however, requires knowledge on transformation of random variables which is detailed below.

A1.3.7 Transformation of Random Variables

Suppose X is a random variable (RV) with known CDF $F_X(x)$ and *pdf* $f_X(x)$ It is often required to find the CDF and *pdf* of a RV Y with a given transformation $Y = g(X)$. The distribution function of the new random variable is given by:

$$F_Y(y) = P(Y \le y) = P(g(X) \le y) = P(X \le g^{-1}(y)) = P(X \in R_y) \qquad \text{(A1.25)}$$

where R_y is the region containing the realizations (samples) x of X for which $g(x) \le y$.

A strictly monotonic functions (either increasing or decreasing) is shown in Figure A1.6. Because of the one-to-one correspondence between X and Y, it is possible to express $P(y \le Y \le y+dy)$ as:

$$P(y \le Y \le y+dy) = P(y \le Y \le y+dy)$$

$$\Rightarrow F_Y(y+dy) - F(y) = F_X(x+dx) - F_X(x)$$

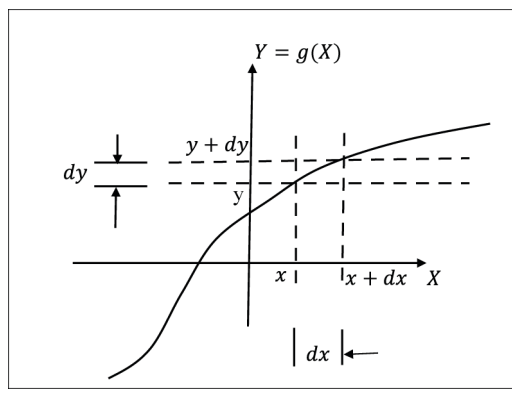


FIGURE A1.6 Transformation of a random variable to another one via a strictly monotonically increasing function.

$$\Rightarrow \mathbb{f}_Y(y)dy = \mathbb{f}_X(x)dx$$

$$\Rightarrow \mathbb{f}_Y(y) = \mathbb{f}_X(x)\frac{dx}{dy} = \mathbb{f}_X(x)\left|\frac{dx}{dy}\right| \tag{A1.26}$$

The $|.|$ sign over $\dfrac{dx}{dy}$ is to render the *pdf* $\mathbb{f}_Y(y)$ always positive. The *pdf* of Y is thus derivable in terms of $x = g^{-1}(y)$ and is given by:

$$\mathbb{f}_Y(y) = \mathbb{f}_X\left(g^{-1}(y)\right)\left|\frac{dx}{dy}\right|_{x=g^{-1}(y)} \tag{A1.27}$$

If $g(X)$ is not a monotonic function, the *pdf* of Y is obtained as:

$$\mathbb{f}_Y(y) = \sum_i \mathbb{f}_X(x_i)\left|\frac{dx}{dy}\right|_{x=x_i} \tag{A1.28}$$

where the summation is over all the real roots $x_i = g^{-1}(y)$. For example, if Y is related to X by $Y = g(X) = X^2$, then:

$$F_Y(y) = P(Y \le y) = P(X^2 \le y) = P\left(X \le \pm\sqrt{y}\right) = P\left(X \in R_y\right)$$

$$\Rightarrow \mathbb{f}_Y(y) = \mathbb{f}_X(x_1)\left|\frac{dx}{dy}\right|_{x=x_1} + \mathbb{f}_X(x_2)\left|\frac{dx}{dy}\right|_{x=x_2} \tag{A1.29}$$

where $x_1 = \sqrt{y}$ and $x = -\sqrt{y}$ are the two possible roots of the transformation (Figure A1.7).

Example A1.1. Given that X is standard normal, find $\mathbb{f}_Y(y)$ where $Y = g(X) = \sigma X + m$. A standard normal is a normal distribution with zero mean and standard deviation equal to unity, i.e., $X \approx N(0,1)$.

Solution. Given $\mathbb{f}_X(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}$, and $x = g^{-1}(y) = \dfrac{y-m}{\sigma}$, Equation (A1.27) gives:

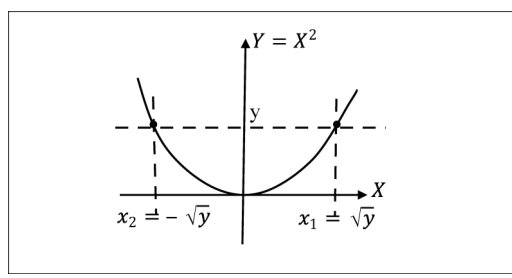


FIGURE A1.7 Transformation of a random variable to another one via a quadratic function.

$$f_Y(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y-m}{a}\right)^2} \tag{A1.30}$$

The linear transformation generated a new normal RV with mean m and with variance = σ^2. Thus, $Y \approx N(m, \sigma)$.

■

Example A1.2. The strain energy in a linear elastic bar subjected to an axial force U is:

$$S = \frac{L}{2AE} F^2 \tag{A1.31}$$

where L is length of the bar, A the area of cross-section of the bar and E the modulus of elasticity of the material. Given $\mathcal{U} \sim N(0,1)$, find the *pdf* $f_S(s)$ of S.

Solution. Given $f_U(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$. From the transformation $S = \frac{L}{2AE}\mathcal{U}^2$, one has: $u = \pm\sqrt{\frac{s}{c}}$, so $u_1 = \sqrt{\frac{s}{c}}$ and $u_2 = -\sqrt{\frac{s}{c}}$ where $c = \frac{L}{2AE}$. From Equation (A1.29),

$$f_S(s) = \frac{f_U\left(\sqrt{\frac{s}{c}}\right)}{\left|\frac{ds}{du}\right|_{u_1}} + \frac{f_U\left(-\sqrt{\frac{s}{c}}\right)}{\left|\frac{ds}{du}\right|_{u_2}} = \frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{s}{2c}}}{2\sqrt{cs}} + \frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{s}{2c}}}{2\sqrt{cs}} \tag{A1.32}$$

with $\frac{ds}{du} = 2cu \Rightarrow \left|\frac{ds}{du}\right| = 2c\sqrt{\frac{s}{c}} = 2\sqrt{cs}$. Therefore:

$$f_S(s) = \frac{1}{\sqrt{2\pi cs}} e^{-\frac{s}{2c}}, \qquad s \geq 0$$

$$= 0, \quad \text{otherwise} \tag{A1.33}$$

■

With a procedure similar to the above case of one-dimensional transformation, a two-dimensional transformation can be also handled. This is illustrated in the following. Suppose that R and θ are two independent RVs with the joint *pdf*:

$$f_{R\theta}(r, \theta) = f_\theta(\theta) f_R(r) = \left(\frac{1}{2\pi}\right)\left(re^{-r^2/2}\right) r \geq 0 \text{ and } 0 \leq \theta \leq 2\pi \tag{A1.34}$$

Note that R and θ are independent RVs with $f_\theta(\theta) = \frac{1}{2\pi}$ being a uniform distribution and $f_R(r) = re^{-r^2/2}$ a Rayleigh distribution. Now, consider the transformation:

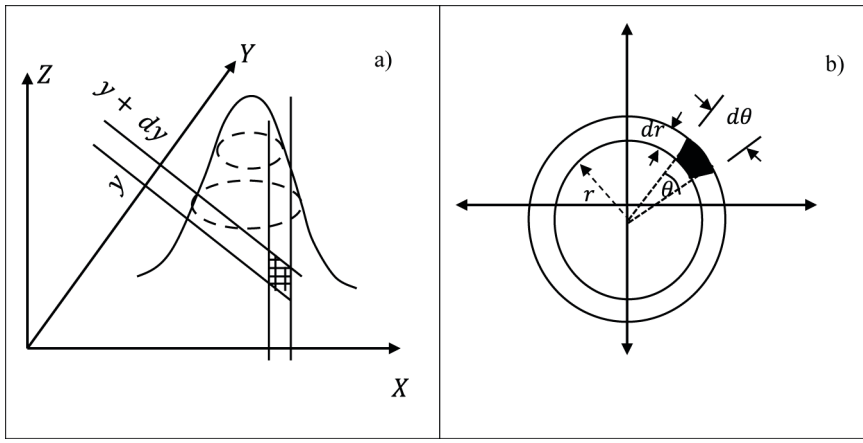


FIGURE A1.8 Two-dimensional transformation; $X = R\cos\theta$, $Y = R\sin\theta$: (a) Cartesian coordinates and (b) polar coordinates.

$$X = R\cos\theta, Y = R\sin\theta \tag{A1.35}$$

The last equation transforms the rotational coordinates R and θ into rectangular coordinates X and Y (Figure A1.8). It is required to find the joint *pdf* $f_{XY}(x,y)$ of X and Y given the *pdf*s of R and θ.

Equating the probability in the infinitesimal volumes (one over the hatched area in Figure A1.8a and the other over the darkened area in Figure A1.8b) corresponding to the two coordinate system, we get $f_{XY}(x,y)dxdy = R\theta(R\theta)drd\theta$.

The area $dxdy$ in the $X-Y$ coordinate system is related to the area $drd\theta$ in $R-\theta$ coordinate system by:

$$dxdy = |J|drd\theta \tag{A1.36}$$

where J is the Jacobian matrix given by:

$$J = \begin{bmatrix} \frac{\partial X}{\partial R} & \frac{\partial X}{\partial \theta} \\ \frac{\partial Y}{\partial R} & \frac{\partial Y}{\partial \theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix} \tag{A1.37}$$

$|J|$ in Equation (A1.37) is the determinant of the matrix J and is equal to r. Therefore, the joint *pdf* of X and Y is given by:

$$\begin{aligned} f_{XY}(x,y) &= \frac{1}{r} f_{R\theta}(R,\theta) \\ &= \frac{1}{r}\left(\frac{1}{2\pi}\right)\left(re^{-r^2/2}\right) \\ &= \frac{1}{2\pi}\, e^{-(x^2+y^2)/2}, -\infty < X < \infty, -\infty < Y < \infty \end{aligned} \tag{A1.38}$$

It is also obvious that the joint *pdf* $f_{XY}(x,y)$ is separable as:

$$f_{XY}(x,y) = \left(\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\right)\left(\frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}\right) \tag{A1.39}$$

Thus, X and Y are also independent RVs. Also, we recognize that both X and Y are standard normals. By the use of the transformation, two independent normal RVs can be generated, once the realizations of the RVs R and θ are available. Samples of the uniformly distributed RV $\theta \approx U(0, 2\pi)$ are available from pseudo random generators (see item 3 of this Appendix) in a computing machine. The Rayleigh RV R may be generated from $U(0, 2\pi)$ by inversion method of MC simulation (Appendix 3). It is now straight forward to get samples of the normal RVs X and Y by the transformation in Equation (A1.11), i.e.:

$$x_i = r_i \cos\theta_i, \text{ and } y_i = r_i \sin\theta_i, i = 1, 2, \ldots \tag{A1.40}$$

The transformation in the above illustration is known as Box-Muller transformation (Box and Muller 1952).

A1.4 LINEAR INDEPENDENCE AND COMPLETENESS

Let $\boldsymbol{Y}_j, j = 1, 2, \ldots, n$ be a sequence of functions in a finite dimensional subspace $\mathbb{Y}_n$ of $\mathbb{Y} \in {}^* C^{n-1}(a, b)$. Consider the following linear combination of these functions:

$$f(x) = a_1 Y_1(x) + a_2 Y_2(x) + \ldots + a_n Y_n(x) \tag{A1.41}$$

If $f(x) = 0$, then its derivatives of orders $1, 2, \ldots, n-1$ are all zero. One can write:

$$\begin{bmatrix} Y_1(x) & Y_2(x) & \ldots & Y_n(x) \\ Y_1^{'}(x) & Y_2^{'}(x) & \ldots & Y_n^{'}(x) \\ & \ldots & & \\ Y_1^{n-1}(x) & Y_2^{n-1}(x) & \ldots & Y_n^{n-1}(x) \end{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ . \\ a_n \end{pmatrix} = 0 \Rightarrow \left[\boldsymbol{W}(x)\right](\boldsymbol{a}) = \boldsymbol{0} \tag{A1.42}$$

If the matrix $\boldsymbol{W}(x)$ is non-singular at any point x in the interval of interest, then necessarily one has $\boldsymbol{a} = \mathbf{0} \Rightarrow a_1 = 0,\ a_2 = 0,\ \ldots,\ a_n = 0$. Then the functions $\boldsymbol{Y}_j, j = 1,2,\ldots,$ are called linearly independent. Note that in the context of ODEs, the matrix $\boldsymbol{W}$ is known as Wronskian.

If $\mathbb{Y} \in \mathcal{H}$, the Hilbert space, the subspace $\mathbb{Y}_n$ is complete if every Cauchy sequence in $\mathbb{Y}_n$ converges to an element in $\mathbb{Y}_n$. For definition of Hilbert space and Cauchy sequence, see Section A1.5.

A1.5 HILBERT SPACE

A complete inner-product space – real or complex – is called a Hilbert space $\mathcal{H}$. Completeness is a property of metric spaces. A space is complete if every Cauchy sequence converges. A sequence $\{x_n\}$ is a Cauchy sequence if for all $\varepsilon > 0$ there exists $N \in \mathbb{N}$ such that for every $m, n > N,\ \|x_n - x_m\| < \varepsilon$ in the sense that the norm of the differences approach zero. Hilbert spaces are named after David Hilbert (1862–1943) who introduced the concept while studying integral equations and functional analysis. All inner product spaces such as Euclidean space associated with the familiar dot product are Hilbert spaces. Specifically, the theory of Hilbert space generalizes the concept of Euclidean space to infinite dimensional space. Examples of Hilbert spaces include spaces of square integrable functions, Sobolev spaces (Reddy 1998) consisting of generalized functions and spaces of sequences.

A1.6 GREEN'S IDENTITY

In finite element method, the given boundary value problem (for instance Equation 1.40 in Chapter 1) is reformulated in an integral form using Green's identity. The integral form is known as weak form in the sense that the required differentiability conditions on the assumed solution (with respect to the spatial coordinates) are less stringent. Green's identity is a combination of the familiar product rule and the divergence theorem.

In vector form, the product rule is:

$$\nabla.(v\boldsymbol{w}) = \nabla v.\boldsymbol{w} + v\nabla.\boldsymbol{w} \tag{A1.43}$$

Here v is a scalar function and $\boldsymbol{w}$ a vector function. By divergence theorem:

$$\int_{\mho} \nabla.(v\boldsymbol{w}) = \int_{\mathrm{S}} (v\vec{n}.\boldsymbol{w})\, d\mathrm{s} \tag{A1.44}$$

where $\vec{n}$ is the outward-directed unit normal on the boundary $d\mho$. Combining Equations (A1.43) and (A1.44) gives:

$$\int_{\mho} \nabla v.\boldsymbol{w} + \int_{\mho} v\nabla.\boldsymbol{w} = \int_{d\mho} (v\vec{n}.\boldsymbol{w})\, d\mathrm{s} \tag{A1.45}$$

If $\boldsymbol{w} = \nabla u$ with u being a scalar function, the last equation leads to Green's identity:

$$\int_{\mho} \nabla v.\nabla u + \int_{\mho} v\nabla.\nabla u = \int_{d\mho} \left(v\vec{n}.\nabla u\right) ds$$

$$\Rightarrow \int_{\mho} v\Delta u = -\int_{\mho} \nabla v.\nabla u + \int_{d\mho} \left(v\frac{\partial u}{\partial n}\right) ds, \text{ where } \Delta = \nabla.\nabla \tag{A1.46}$$

$\frac{\partial u}{\partial n} = \vec{n}.\nabla u$ is the directional derivative of the function u in the direction of the unit normal. In the example problem of Figure 1.19, the test function U_j and trial function Y_j stand respectively for v and u in Equation (A1.46). The length of the rod $0-l$ defines the domain $\mho$ and the boundary $d\mho$ consists of the two end points of the rod. Therefore, the weak form for the vibrating rod is obtained from Equation (1.42) in Chapter 1 as:

$$\int_0^l U_j(x)\left\{\mathcal{A}(\tilde{y})\right\} d\boldsymbol{x} = \int_0^l U_j(x) f_A dx, \quad j = 1,2,\ldots$$

$$\int_0^l U_j(x)\left\{-\frac{\partial}{\partial x}\left(EA\frac{\partial y}{\partial x}\right) + m\frac{\partial^2 y}{\partial t^2}\right\} d\boldsymbol{x} = \int_0^l U_j(x) f_A dx, \quad j = 1,2,\ldots \tag{A1.47a}$$

From Equation (1.41) in Chapter 1, one has:

$$\sum_j \left[-q_j(t)\int_0^l U_j(x)\left\{\frac{\partial}{\partial x}\left(EA\frac{\partial Y_j}{\partial x}\right)\right\} d\boldsymbol{x} + m\ddot{q}_j \int_0^l U_j(x) Y_j(x) dx \right] = \int_0^l U_j(x) f_A dx, \quad j = 1,2,\ldots \tag{A1.47b}$$

Application of Green's identity to the first integral on LHS of Equation (A1.47b) gives:

$$\sum_j \left\{ q_j(t)\left\{ \int_0^l \frac{dU_j(x)}{dx}\left(EA\frac{dY_j(x)}{dx}\right) dx - U_j\frac{dY_j}{dn}\Big|_0^l \right\} + m\ddot{q}_j\int_0^l U_j(x) Y_j(x) d\boldsymbol{x}\right\} a = \int_0^l U_j(x) f_A dx, \quad j = 1,2,\ldots \tag{A1.48}$$

The last equation is identical to Equation (1.43) in Chapter 1. Note that the boundary condition term $U_j \left.\frac{\partial Y_j}{\partial n}\right|_0^l$ naturally emerges out of the weak formulation.

A1.7 BILINEAR FORM ON $\mathcal{H}\times\mathcal{H}$ AND LINEAR FORM ON $\mathcal{H}$

$\alpha(.,.)$ is a bilinear form on $V\times V$ if $\alpha: V\times V\to\mathbb{R}$, i.e., $\alpha(u,v)\in\mathbb{R}$ for all $u,v\in V$ and it is linear in both the arguments:

$$\alpha(au+bw,v)=a\alpha(u,v)+b\alpha(w,v)$$

$$\alpha(u,aw+bv)=a\alpha(u,w)+b\alpha(u,v),$$
$$\forall u,v,w\in V \quad \text{and} \quad a,b\in\mathbb{R} \tag{A1.49}$$

A bilinear form $\alpha(.,.)$ on $V\times V$ is symmetric if:

$$\alpha(u,v)=\alpha(v,u), \quad \forall u,v\in V \tag{A1.50}$$

A symmetric bilinear form is positive semi definite if $\alpha(u,u)\geq 0,\ \forall u\in V$ and positive definite if $(u,u)>0,\ \forall u\in V$. A symmetric bilinear form $\alpha(u,v): V\times V\to\mathbb{R}$ is an inner product on V if, for $\forall u, v\in V$:

$$\alpha(u,u)\geq 0 \quad \text{and} \quad \alpha(u,u)=0\Rightarrow u=0 \tag{A1.51}$$

The norm associated with the inner product is defined by:

$$\|u\|_V=\left(\alpha(u,u)\right)^{1/2}, \quad \forall v\in V \tag{A1.52}$$

$\ell(.)$ is a linear form on V if $\ell: V\to\mathbb{R}$, i.e., $l(u)\in\mathbb{R}$ for $u\in V$. It is linear, i.e., for all $u,v\in V$, we have:

$$\ell(au+bv)=a\ell(u)+b\ell(v),\ a,b\in\mathbb{R} \tag{A1.53}$$

A1.8 WEAK DERIVATIVE OF A FUNCTION IN $\mathcal{H}$, THE HILBERT SPACE

Let $\mathfrak{V}$ be a domain in $\mathbb{R}^n$ and $D(\mathfrak{V})$ the set of $C^\infty(\mathfrak{V})$* functions with compact support in $\mathfrak{V}$. Let us also define the set of locally integrable functions in the compact $K\subset\mathfrak{V}$:

$$L^1_{loc}(\mathfrak{V}):=\left\{u:\ u\in L^1_K \quad \forall K\subset\mathfrak{V}\right\} \tag{A1.54}$$

* $C^m(a, b)$ functions
$y(x)$ is called a $C^m(a, b)$ function if its derivative of all orders ≤ m – 1 exist and are continuous over the interval [a, b].

L_K^1 denotes a locally integrable function such that for any $u \in L_K^1$, the integral $\int_K |u(x)| dx$ is finite on every compact subset $K \subset \mho$. Now, we say a given function $u \in L_{loc}^1(\mho)$ has a weak derivative $D^\alpha u$, provided there exists a function $D^\alpha u \in L_{loc}^1(\mho)$ such that:

$$\int_{\mho} D^\alpha u(x)\,(x)\,dx = (-1)^{|\alpha|} \int_{\mho} u(x)\; D^\alpha \phi(x)\; dx, \forall \phi \in D(\mho) \tag{A1.55}$$

If such a function exists we say that $D^\alpha u$ is the weak derivative of $u(x)$. Here α is a multi-index and is an ordered n-tuples of non-negative integers. $|\alpha|$ is a notation for the sum $\alpha_1 + \alpha_1 + \ldots + \alpha_n$ with α_i being a non-negative integer. $D^\alpha u$ denote the partial derivative:

$$\mathrm{D}^\alpha \mathrm{u} = \frac{\partial^{|\alpha|} \mathrm{u}}{\partial \mathrm{x}_1^{\alpha_1} \partial \mathrm{x}_2^{\alpha_2} \ldots \partial \mathrm{x}_\mathrm{n}^{\alpha_\mathrm{n}}} \tag{A1.56}$$

Thus if $|\alpha| = m$, $D^\alpha u$ is one of the m^{th} partial derivatives of u. If $n = 3$, we can take, for example, $\alpha = (1,0,3)$. With $|\alpha| = 1 + 0 + 3 = 4$, $D^\alpha u$ is the fourth order partial derivative given by:

$$D^\alpha u = \frac{\partial^4 u}{\partial x_1^1 \partial x_2^0 \partial x_n^3} = \frac{\partial^4 u}{\partial x_1 \partial x_3^3} \tag{A1.57}$$

See Reddy (1998) for further details.

A1.9 FARKAS'S LEMMA

Equation (1.61a) in Chapter 1 is the KKT condition for a constrained optimization problem and if it is satisfied, the optimum is realized and no further search is possible for a direction $\boldsymbol{d}$ which is both descent and feasible. This implies that the intersection of descent and feasible cones is empty (Figure 1.26), i.e., Equation (1.61a) in Chapter 1 has no solution and $F \cap H \cap G = \phi$. See Section 1.6.3, Chapter 1, for definitions of F, G and H. This is highlighted by Farkas' lemma. The statement of the lemma is as follows:

Let $\boldsymbol{A} \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}}$ and $\boldsymbol{b} \in \mathbb{R}^{\mathrm{n}}$. Then, only one of the following two systems holds but not both.

(i) $\exists\, \boldsymbol{y} \in \mathbb{R}^m$ such that $\boldsymbol{A}^T \boldsymbol{y} = \boldsymbol{b}$ and $\boldsymbol{y} \geq 0$

(ii) $\exists \boldsymbol{z} \in \mathbb{R}^n$ such that $\boldsymbol{A}\boldsymbol{z} \geq 0$ and $\boldsymbol{z}^T \boldsymbol{b} < 0$ (A1.58)

In the context of KKT optimality conditions, rows of the matrix $\boldsymbol{A}$ in the two systems represents the n-dimensional gradient vectors $-\nabla t_i(\boldsymbol{x})$ of the constraints

$t_i(\boldsymbol{x}), i = 1,2,\ldots,m$ (equality or inequality) with $\boldsymbol{x} \in \mathbb{R}^n$. The vector $\boldsymbol{b}$ represents ∇f, the gradient of the objective function. That the system (i) holds implies that the KKT conditions $-\nabla f(\boldsymbol{x}^*) = \sum_i^m \lambda_i t_i(\boldsymbol{x}^*)$ (corresponding to Equation 1.49a of equality constraint or 1.53a of inequality constraint in Chapter 1) with $\lambda_i \geq 0$ are satisfied with at least a few of the constraints being active, i.e., $t_i(\boldsymbol{x}^*) = 0$ and the associated λ_i being strictly positive. The vector $\boldsymbol{y}$ in system (i) stands for the vector of Lagrangian multipliers $\lambda_i, i = 1,2,\ldots,m$. Suppose that the above KKT conditions are not satisfied by $\boldsymbol{x}$, i.e., the system (i) fails to hold, i.e., optimum is not yet realized. Noting that the vector $\boldsymbol{z}$ in the system (ii) represents $\boldsymbol{d}$, it is possible to have a search direction $\boldsymbol{d} \in \mathbb{R}^n$ which is a descent direction $\nabla f^T \boldsymbol{d} < 0$ and also a feasible one, i.e., $-\nabla t^T \boldsymbol{d} \geq 0$. This corresponds to the system (ii). See Bazaraa *et al*. 2006 for further details on Farkas's lemma.

A1.10 SADDLE POINT

To decide the behaviour of an objective function $L(\boldsymbol{x}^*)$ at a stationary point, it is needed to examine the Hessian $\boldsymbol{H}(\boldsymbol{x}^*) = \nabla^2 L(\boldsymbol{x})$. Note that $\boldsymbol{H}$ is symmetric. If it is positive definite, $\boldsymbol{x}^*$ is a minimum point and if negative definite, $\boldsymbol{x}$ is a maximum point. If $\boldsymbol{H}(\boldsymbol{x}^*)$ is indefinite, $\boldsymbol{x}^*$ is a saddle point. While the eigenvalues of a positive definite symmetric $\boldsymbol{H}(\boldsymbol{x}^*)$ are all positive and are all negative for a negative definite $\boldsymbol{H}(\boldsymbol{x}^*)$, an indefinite $\boldsymbol{H}(\boldsymbol{x}^*)$ possess both positive and negative eigenvalues. Since $\boldsymbol{H}(\boldsymbol{x}^*)$ signifies the curvature of the function at $\boldsymbol{x}^*$, the positive and negative eigenvalues correspond to the extreme values of the curvature of the function in different directions – one of positive curvature and the other of negative curvature.

For instance, $\boldsymbol{H}(\boldsymbol{x})$ of the objective function $L(x,y) = x^2 + y^2 + rxy - 2x$, $r \in \mathbb{R}$ is

$$\boldsymbol{H}(\boldsymbol{x}) = \begin{bmatrix} 2 & r \\ r & 2 \end{bmatrix} \tag{A1.59}$$

In terms of *r*, the optimum point $\boldsymbol{x}^* = \left(\frac{2}{4-r^2}, \frac{-2r}{4-r^2} \right)$ With $r < 2$, both eigenvalues are positive and $\boldsymbol{x}^*$ is minimum. For $r > 2$, the eigenvalue are of opposite sign and $\boldsymbol{x}^*$ is a saddle point. Suppose if $r = 3$, $\boldsymbol{x}^* = (-0.4, 1.2)^T$ and the eigenvalues of $\boldsymbol{H}(\boldsymbol{x})$ are –1 and 5. With $r = 2$, there is no stationary point.

A1.11 LEGENDRE TRANSFORM

Suppose that $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ with $\boldsymbol{x} \in \mathbb{R}^n$ is a convex differentiable function. In terms of a variable $\boldsymbol{p} \in \mathbb{R}^n$ conjugate to $\boldsymbol{x}$, the Legendre transform denoted by $\mathcal{T}(f(\boldsymbol{x})) : \mathbb{R}^n \to \mathbb{R}$ is given by:

$$\mathcal{T}\left(f\left(\boldsymbol{x}\right)\right)=h\left(\boldsymbol{p}\right)=\max_{x}\left[\boldsymbol{xp}-f\left(\boldsymbol{x}\right)\right] \tag{A1.60}$$

$\boldsymbol{xp}$ in the above equation means a vector dot product $\boldsymbol{x}\cdot\boldsymbol{p}$. Also, $max_{x}\left[.\right]$ indicates maximization of the expression within the square brackets, with respect to $\boldsymbol{x}$ whilst $\boldsymbol{p}$ is held constant. Thus, $\boldsymbol{p}=f'\left(\boldsymbol{x}_{p}\right)$ and $\boldsymbol{x}_{p}\in\mathbb{R}^{n}$ is the point where the bracketed expression is a maximum. The expression $\boldsymbol{xp}-f\left(\boldsymbol{x}\right)$ is maximized since $f\left(\boldsymbol{x}\right)$ is convex and the second order derivative $\frac{d^{2}}{d\boldsymbol{x}^{2}}\left(\boldsymbol{xp}-\boldsymbol{f}\left(\boldsymbol{x}\right)\right)=-f''\left(\boldsymbol{x}\right)$ is a negative definite $n\times n$ matrix. One has:

$$h\left(\boldsymbol{p}\right)=\boldsymbol{x}_{p}\boldsymbol{p}-f\left(\boldsymbol{x}_{p}\right) \tag{A1.61}$$

Example. Consider the EL equations with respect to Lagrangian $L\left(x,\dot{x}\right)$ (see Equation 1.18 in Chapter 1):

$$\frac{\partial L}{\partial x}-\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{x}}\right)=0 \tag{A1.62}$$

Defining p conjugate to $\dot{x}$, let us transform the Lagrangian $L\left(x,\dot{x}\right)$ to the Hamiltonian $H\left(x,p\right)=\mathcal{T}\left(L\right)$ by Legendre transform. From Equations (A1.60–61), $p=\frac{\partial L}{\partial\dot{x}}\Rightarrow\dot{x}=\phi\left(x,p\right)$, say, and $H\left(x,p\right)=p\dot{x}-L\left(x,\dot{x}\right)=p\phi\left(x,p\right)-L\left(x,\phi\left(x,p\right)\right)$.
Let us now show that the canonical Hamiltonian equations are:

$$\dot{x}=\frac{\partial H}{\partial p}\text{ and }\dot{p}=-\frac{\partial H}{\partial x} \tag{A1.63–A1.64}$$

First consider the RHS of Equation (A1.63).

$$\frac{\partial H}{\partial p}=\frac{\partial}{\partial p}\left\{p\phi\left(x,p\right)-L\left(x,\phi\left(x,p\right)\right)\right\}$$

$$=\phi\left(x,p\right)-p\frac{\partial\phi\left(x,p\right)}{\partial p}-\frac{\partial L\left(x,\phi\left(x,p\right)\right)}{\partial\phi\left(x,p\right)}\frac{\partial\phi\left(x,p\right)}{\partial p}$$

$=\phi\left(x,p\right)$ (since the last two terms on the extreme RHS cancel with each other)

$$=\dot{x}=\text{LHS of Equation}\left(\text{A1.63}\right) \tag{A1.65}$$

Now from the RHS of Equation (A1.64), one has:

$$\frac{\partial H}{\partial x} = \frac{\partial}{\partial x}\left\{p\phi(x,p) - L\left(x,\phi(x,p)\right)\right\}$$
$$= p\frac{\partial \phi(x,p)}{\partial x} - \frac{\partial L\left(x,\phi(x,p)\right)}{\partial x} - \frac{\partial L\left(x,\phi(x,p)\right)}{\partial \phi(x,p)}\frac{\partial \phi(x,p)}{\partial x}$$

$$= -\frac{\partial L(x,\dot{x})}{\partial x}\left(\text{since}\frac{\partial \phi(x,p)}{\partial x} = 0\right) \tag{A1.66}$$

But from EL equation, $\frac{\partial L(x,\dot{x})}{\partial x} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right)$. So, from Equation (A1.66):

$$\frac{\partial H}{\partial x} = -\frac{\partial L(x,\dot{x})}{\partial x} = -\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = -\dot{p} = \text{LHS of Equation}\left(\text{A1.64}\right) \tag{A1.67}$$

The change of coordinates from $(x,\dot{x})$ to (x,p) gives the Jacobian $= \det\begin{bmatrix} 1 & 0 \\ \frac{\partial p}{\partial x} & \frac{\partial p}{\partial \dot{x}} \end{bmatrix} = \frac{\partial p}{\partial \dot{x}} = \frac{\partial^2 L}{\partial \dot{x}^2}$ which shows that the Legendre transformation is invertible if $\frac{\partial^2 L}{\partial \dot{x}^2} \neq 0$.

A1.12 BELLMAN PRINCIPLE OF OPTIMALITY (BELLMAN AND KALABA 1964) AND DERIVATION OF THE HAMILTON-JACOBI-BELLMAN (HJB) EQUATION

Consider an n-dimensional dynamical system as described by Equation 1.71b in Chapter 1 (Section 1.7), reproduced below for a ready reference.

$$\dot{\boldsymbol{x}} = \boldsymbol{F}\left(\boldsymbol{x}(t),\boldsymbol{u}(t),t\right) \tag{A1.68}$$

The objective is presently to minimize a cost functional J with respect to the control variable $\boldsymbol{u}(t)$ as:

$$\text{minimize } J\left(\boldsymbol{x}_0,\boldsymbol{u},t_0\right) = h\left(\boldsymbol{x}(T)\right) + \int_{t_0}^{T} L\left(\boldsymbol{x}(t),\boldsymbol{u}(t),t\right)dt$$

$$\text{s.t.} x(t) \text{ satisfies Equation}\left(\text{A1.68}\right) \tag{A1.69}$$

Here $L: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ is a smooth convex function referred to as the running cost and $h: \mathbb{R}^n \to \mathbb{R}$ is the terminal cost. m is the dimension of $\boldsymbol{u}(t)$. $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$ is the initial state. $\boldsymbol{F}: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^n$ is a vector-valued function defining the system dynamics. T is the terminal time. With $\boldsymbol{x}(t) := \boldsymbol{x}_t$ and $\boldsymbol{u}(t) := \boldsymbol{u}_t$ we denote the set of all admissible control functions by $\boldsymbol{U}(0,T)$. To proceed further, one typically defines a value function $V(\boldsymbol{x}_t, t)$ as the minimal cost to reach some final state $\boldsymbol{x}_T$, given the starting point at $\boldsymbol{x}_t$. That is:

$$\begin{aligned} V(\boldsymbol{x}_t, t) &= \min_{\substack{\boldsymbol{u}_\tau \in U \\ t \le \tau \le T}} J(\boldsymbol{x}_t, \boldsymbol{u}_t, t) \\ &= \min_{\substack{\boldsymbol{u}_\tau \in U \\ t \le \tau \le T}} \left(h(\boldsymbol{x}(T)) + \int_t^T L(\boldsymbol{x}_\tau, \boldsymbol{u}_\tau, \tau) d\tau \right) \end{aligned} \tag{A1.70}$$

The optimality principle may be restated in a recursive form by restricting to an interval $(t, t+\Delta t)$. To this end, we first split the time interval (t,T) as $(t, t+\Delta t] \cup [t+\Delta t, T)$, Equation (A1.70) may thus be written as:

$$\begin{aligned} V(\boldsymbol{x}_t, t) = \min_{\substack{\boldsymbol{u}_\tau \in U \\ t \le \tau \le T}} \Bigg(& h(\boldsymbol{x}_T) + \int_t^{t+\Delta t} L(\boldsymbol{x}_\tau, \boldsymbol{u}_\tau, \tau) d\tau \\ & + \int_{t+\Delta t}^T L(\boldsymbol{x}_\tau, \boldsymbol{u}_\tau, \tau) d\tau \Bigg) \end{aligned} \tag{A1.71}$$

Recognizing $\min_{\substack{u_\tau \in U \\ t \le \tau \le T}} \left(h(\boldsymbol{x}_T) + \int_{t+\Delta t}^T L(\boldsymbol{x}_\tau, \boldsymbol{u}_\tau, \tau) d\tau \right)$ *τ, and T* as $V(\boldsymbol{x}_{t+\Delta t}, t+\Delta t)$, we have:

$$V(\boldsymbol{x}_t, t) = \min_{\substack{\boldsymbol{u}_\tau \in U \\ t \le \tau \le T}} \left(\int_t^{t+\Delta t} L(\boldsymbol{x}_\tau, \boldsymbol{u}_\tau, \tau) d\tau + V(\boldsymbol{x}_{t+\Delta t}, t+\Delta t) \right) \tag{A1.72}$$

Further, as Δt is small, one may write $\int_t^{t+\Delta t} L(\boldsymbol{x}_\tau, \boldsymbol{u}_\tau, \tau) dt \cong L(\boldsymbol{x}_t, \boldsymbol{u}_t, t) \Delta t$ and hence get the value function in a recursive form:

$$V(\boldsymbol{x}_t, t) \cong \min_{u_t \in U} \left\{ L(\boldsymbol{x}_t, \boldsymbol{u}_t, t) \Delta t + V(\boldsymbol{x}_{t+\Delta t}, t+\Delta t) \right\} \tag{A1.73}$$

Expanding $V\left(\boldsymbol{x}_{t+\Delta t}, t+\Delta t\right)$ by Taylor's expansion about $V\left(\boldsymbol{x}_t, t\right)$ and neglecting the terms quadratic or still higher in Δt and $\Delta \boldsymbol{x}$, Equation (A1.73) takes the form:

$$V\left(\boldsymbol{x}_t, t\right) \cong \min_{\boldsymbol{u}_t \in U}\left\{L\left(\boldsymbol{x}_t, \boldsymbol{u}_t, t\right)\Delta t + \left(V\left(\boldsymbol{x}_t, t\right) + \Delta t\left(\frac{\partial V}{\partial t} + \left(\frac{\partial V}{\partial x}\right)^T \dot{\boldsymbol{x}}_t\right)\right)\right\} \tag{A1.74}$$

With cancellation of $V\left(\boldsymbol{x}_t, t\right)$ from both sides (note that the term is independent of the control variable $\boldsymbol{u}(t)$) and division by Δt, we take limit as $\Delta t \to 0$ to yield:

$$0 = \frac{\partial V}{\partial t} + \min_{\boldsymbol{u}_t \in U}\left\{L\left(\boldsymbol{x}_t, \boldsymbol{u}_t, t\right) + \left(\frac{\partial V}{\partial x}\right)^T \boldsymbol{F}\left(\boldsymbol{x}_t, \boldsymbol{u}_t, t\right)\right\} \tag{A1.75}$$

Equation (A1.51) is the HJB equation which is a nonlinear PDE. The solution to the HJB equation gives the value function V which, being the optimal cost of the control problem, yields the control variable $\boldsymbol{u}(t)$.

A1.12.1 LQR Problem (Deterministic Case) and HJB Equation

If it is an LQR problem as in Example 1.5 in Chapter 1, we have $\boldsymbol{F}\left(\boldsymbol{x}_t, \boldsymbol{u}_t, t\right) = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t$ and $L\left(\boldsymbol{x}, \boldsymbol{u}, t\right) = \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}$. See Equations 1.97–1.98 in Chapter 1 for definitions of $\boldsymbol{Q}, \boldsymbol{R}, \boldsymbol{A}$ and $\boldsymbol{B}$ which are constant matrices. Equation (A1.51) then takes the form:

$$0 = \frac{\partial V}{\partial t} + \min_{\boldsymbol{u}_t \in U}\left(\boldsymbol{x}_t^T\boldsymbol{Q}\boldsymbol{x}_t + \boldsymbol{u}_t^T\boldsymbol{R}\boldsymbol{u}_t\right) + \left(\frac{\partial V}{\partial x}\right)^T\left(\boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t\right) \tag{A1.76}$$

For the HJB equation (A1.76), one may try a solution in the form $V\left(\boldsymbol{x}_t, t\right) = \boldsymbol{x}_t^T\boldsymbol{K}_t\boldsymbol{x}_t$ with $\boldsymbol{K}$ symmetric. Using $\frac{\partial V}{\partial t} = \boldsymbol{x}_t^T\dot{\boldsymbol{K}}_t\boldsymbol{x}_t$ and $\frac{\partial V}{\partial x} = 2\boldsymbol{K}_t\boldsymbol{x}_t$, we substitute the assumed solution for V in Equation (A1.76), and apply the first order optimality condition $\frac{dV}{d\boldsymbol{u}} = 0$ to get:

$$2\boldsymbol{B}^T\boldsymbol{K}_t\boldsymbol{x}_t + 2\boldsymbol{R}\boldsymbol{u}_t = 0 \Rightarrow \boldsymbol{u}_t = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{K}_t\boldsymbol{x}_t \tag{A1.77}$$

Thus, we find that the solution to the control variable $\boldsymbol{u}_t$ in terms of $\boldsymbol{x}_t$ is the same as the one obtained in Chapter 1 by the Pontryagin's minimum principle (see Equation 1.92 of Chapter 1). Substituting $\boldsymbol{u}_t$ in the HJB equation (A1.76) gives:

$$\dot{\boldsymbol{K}}_t = -\boldsymbol{Q} - \boldsymbol{A}^T \boldsymbol{K}_t - \boldsymbol{K}_t \boldsymbol{A} + \boldsymbol{K}_t \boldsymbol{B} \boldsymbol{R}^{-} \boldsymbol{B}^T \boldsymbol{K}_t \tag{A1.78}$$

Note that Equation (A1.76) is the Riccati Equation (1.94) in Chapter 1 which is a nonlinear ODE with a terminal condition $\boldsymbol{K}\left(t_f\right) = \boldsymbol{S}$ (see also Equation 1.95 in Chapter 1). The equation needs to be solved backwards while the system dynamical equation $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t$ is solved in forward time with the initial condition $\boldsymbol{x}\left(t_0\right) = \boldsymbol{x}_0$. As may be observed, for the LQR problem, the two ODEs are uncoupled and the solution may be obtained with little difficulty. However, for a general nonlinear dynamical system, the HJB PDE (A1.75) is not amenable for an easy solution and the same holds for the SOC problem.

REFERENCES

Bazaraa, M. S., H. D. Sherali, and C. M. Shetty. 2006. *Nonlinear programming*. NJ: John Wiley & Sons. Inc.

Bellman, R. E. and R. E. Kalaba. 1964. *Quasilinearization*. New York: Elsevier.

Box, G. E. P. and M. E. Muller. 1958. A note on generation of random normal variates. *The Annals of Mathematical Statistics* 29(2): 610–611.

Garey, M. R. and D. S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP Completeness.* New York: W. H. Freeman & Co.

Knuth, D. E. 1997. *The art of Computer Programming. Volume 1. Fundamental algorithms. 3rd Ed*. Addison and Wesley.

L'Ecuyer, P. 1992. Testing random number generators. Proceedings of the 1992 Winter Simulation Conference (IEEE Press), 305–313.

Lance, F. 2009. The status of the P versus NP problem. *Communications of the ACM* 52(9): 78–86.

Leeuwen, Jan van, ed. 1998. *Handbook of Theoretical Computer Science. vol. A1. Algorithms and Complexity.* Amsterdam: Elsevier.

Markowitz, H. M. 1952. Portfolio selection. *Journal of Finance*, 7(1): 77–91.

Martin, J. C. 1997. *Introduction to Languages and the Theory of Computation,* 2nd ed. New York: McGraw-Hill.

Nishimura, T. 2000. Tables of 64-bit mersenne twisters. *ACM Transactions on fModeling and Computer Simulation*, 10(4): 348–357.

Papoulis, A. 1991. *Probability, Random Variables and Stochastic Processes*. 3rd Ed. New York: McGraw-Hill.

Reddy, D. 1998. *Introductory Functional Analysis with Applications to Boundary Value Problems and Finite Elements*. New York: Springer-Verlag.

Roy, D. and G. V. Rao. 2017. *Stochastic Dynamics, Filtering and Optimization*. Cambridge: Cambridge University Press.

Appendix 2

A2.1 SOBOLEV SPACE

A Sobolev space of order m, denoted by $H^m(\Omega)$ is defined to be the space that consists of functions in $L^2(\Omega)$ that together with all their weak derivatives up to and including m, belong to $L^2(\Omega)$. $L^2(\Omega)$ is the space of square integrable functions on Ω as:

$$L^2(\Omega) = \left\{ v: v \text{ is defined on } \Omega \text{ such that} \int_\Omega v^2 dx < \infty \right\} \tag{A2.1}$$

See Appendix 1 for the definition of a weak derivative. Thus:

$$H^m(\Omega) = i\left\{ v: D^\alpha v \in L^2(\Omega), \text{ for all } \pm \text{ such that } |\alpha| \le m \right\} \tag{A2.2}$$

where α is called the multi-index and $D^\alpha v$ denotes the weak derivative of v. $H^m(\Omega)$ is a Hilbert space (Appendix 1) with an inner product defined by:

$$(u,v)_{H^m} = \int_\Omega \sum_{\alpha \le m} D^\alpha u \, D^\alpha v \, dx \text{ for } u, v \in H^m(\Omega) \tag{A2.3}$$

A2.2 STIFFNESS MATRIX, $\boldsymbol{K}^e$ and the Sensitivity Matrix, $\frac{\partial \boldsymbol{K}^e}{\partial x_i}, i = 1, 2, \ldots 10$

Each element stiffness matrix $\left[\boldsymbol{K}^e\right]_{n\times n}$ is symmetric where $n = 8$ is the number of *dof*s of the truss element (Figure 2.12). The sensitivity matrix $\frac{\partial \boldsymbol{K}^e}{\partial x_i}, i = 1,2,\ldots,10$ for each element is also symmetric.

Initialize $\left[\boldsymbol{K}^e\right]_{8x8}=\left[\boldsymbol{0}\right]_{8x8}$ and $\left[\frac{\partial K^e}{\partial x_i}\right]_{8x8}=\left[0\right]_{8x8}$, for $i=1,2,\ldots,10$. Let $c = \cos^2(\pi/4)/sqrt(2)$. The non-zero elements of $\boldsymbol{K}^e$ above its diagonal are:

$$K^e(1,1)=\frac{E}{L}\left(A_1+A_2+cA_8+cA_{10}\right),\ K^e(1,2)=\frac{E}{L}\left(-cA_8+cA_{10}\right),$$

$$K^e(1,3)=-\frac{E}{L}A_2,\ K^e(1,7)=-c\frac{E}{L}A_{10},\ K^e(1,8)=-c\frac{E}{L}A_{10},$$

$$K^e(2,2)=\frac{E}{L}\left(A_5+cA_8+cA_{10}\right),\ K^e(2,6)=-\frac{E}{L}A_5,\ K^e(2,7)=-c\frac{E}{L}A_{10},$$

$$K^e(2,8)=-c\frac{E}{L}A_{10},$$

$$K^e(3,3)=\frac{\boldsymbol{E}}{\boldsymbol{L}}\left(\boldsymbol{A}_2+c\boldsymbol{A}_9\right),\ K^e(3,4)=-c\frac{\boldsymbol{E}}{\boldsymbol{L}}\boldsymbol{A}_9,\ K^e(3,5)=-c\frac{\boldsymbol{E}}{\boldsymbol{L}}\boldsymbol{A}_9,$$

$$K^e(3,6)=c\frac{\boldsymbol{E}}{\boldsymbol{L}}\boldsymbol{A}_9,$$

$$K^e(4,4)=\frac{\boldsymbol{E}}{\boldsymbol{L}}\left(\boldsymbol{A}_6+c\boldsymbol{A}_9\right),\ K^e(4,5)=c\frac{\boldsymbol{E}}{\boldsymbol{L}}\boldsymbol{A}_9,\ K^e(4,6)=-c\frac{\boldsymbol{E}}{\boldsymbol{L}}\boldsymbol{A}_9,\ K^e(4,8)=-\frac{\boldsymbol{E}}{\boldsymbol{L}}\boldsymbol{A}_6,$$

$$K^e(5,5)=\frac{E}{L}\left(\boldsymbol{A}_3+A_4+cA_7+cA_9\right),\ K^e(5,6)=c\frac{E}{L}\left(A_7-A_9\right),\ K^e(5,7)=-\frac{E}{L}A_4,$$

$$K^e(6,6)=\frac{\boldsymbol{E}}{\boldsymbol{L}}\left(\boldsymbol{A}_5+c\boldsymbol{A}_7+c\boldsymbol{A}_9\right),\ K^e(7,7)=\frac{\boldsymbol{E}}{\boldsymbol{L}}\left(\boldsymbol{A}_4+c\boldsymbol{A}_{10}\right),\ K^e(7,8)=c\frac{\boldsymbol{E}}{\boldsymbol{L}}\boldsymbol{A}_{10},$$

$$K^e(8,8)=\frac{\boldsymbol{E}}{\boldsymbol{L}}\left(\boldsymbol{A}_6+c\boldsymbol{A}_{10}\right) \tag{A2.4}$$

With the element stiffness matrix, the non-zero elements of $\frac{\partial \boldsymbol{K}}{\partial \boldsymbol{x}_i}, i=1,2,\ldots,10$ are obtained as follows:

$$\frac{\partial K^e}{\partial x_1}(1,1)=\frac{E}{L},$$

$$\frac{\partial K^e}{\partial x_2}(1,1)=\frac{E}{L},\ \frac{\partial K^e}{\partial x_2}(1,3)=-\frac{E}{L},\ \frac{\partial K^e}{\partial x_2}(3,3)=\frac{E}{L},\ \frac{\partial K^e}{\partial x_3}(5,5)=\frac{E}{L},$$

$$\frac{\partial K^e}{\partial x_4}(5,5)=\frac{E}{L},\ \frac{\partial K^e}{\partial x_4}(5,7)=-\frac{E}{L},\ \frac{\partial K^e}{\partial x_4}(7,7)=\frac{E}{L},\ \frac{\partial K^e}{\partial x_5}(2,2)$$
$$=\frac{E}{L},\ \frac{\partial K^e}{\partial x_5}(2,6)=-\frac{E}{L},$$

$$\frac{\partial K^e}{\partial x_5}(6,6)=\frac{E}{L},\ \frac{\partial K^e}{\partial x_6}(4,4)=\frac{E}{L},\ \frac{\partial K^e}{\partial x_6}(4,8)=-\frac{E}{L},\ \frac{\partial K^e}{\partial x_6}(8,8)$$
$$=\frac{E}{L},\ \frac{\partial K^e}{\partial x_7}(5,5)=c\frac{E}{L},$$

$$\frac{\partial K^e}{\partial x_7}(5,6)=c\frac{E}{L},\ \frac{\partial K^e}{\partial x_7}(6,6)=c\frac{E}{L},\ \frac{\partial K^e}{\partial x_8}(1,1)=c\frac{E}{L},\ \frac{\partial K^e}{\partial x_8}(1,2)$$
$$=-c\frac{E}{L},\ \frac{\partial K^e}{\partial x_8}(2,2)=c\frac{E}{L},$$

$$\frac{\partial K^e}{\partial x_9}(3,3)=c\frac{E}{L},\ \frac{\partial K^e}{\partial x_9}(3,4)=-c\frac{E}{L},\ \frac{\partial K^e}{\partial x_9}(3,5)=-c\frac{E}{L},\ \frac{\partial K^e}{\partial x_9}(3,6)$$
$$=c\frac{E}{L},\ \frac{\partial K^e}{\partial x_9}(4,4)=c\frac{E}{L},$$

$$\frac{\partial K^e}{\partial x_9}(4,5)=c\frac{E}{L},\ \frac{\partial K^e}{\partial x_9}(4,6)=-c\frac{E}{L},\ \frac{\partial K^e}{\partial x_9}(5,5)=c\frac{E}{L},\ \frac{\partial K^e}{\partial x_9}(5,6)$$
$$=-c\frac{E}{L},\ \frac{\partial K^e}{\partial x_9}(6,6)=c\frac{E}{L},$$

$$\frac{\partial K^e}{\partial x_{10}}(1,1)=c\frac{E}{L},\ \frac{\partial K^e}{\partial x_{10}}(1,2)=c\frac{E}{L},\ \frac{\partial K^e}{\partial x_{10}}(1,7)=-c\frac{E}{L},\ \frac{\partial K^e}{\partial x_{10}}(1,8)$$
$$=-c\frac{E}{L},\ \frac{\partial K^e}{\partial x_{10}}(2,2)=c\frac{E}{L},$$

$$\frac{\partial K^e}{\partial x_{10}}(2,7) = -c\frac{E}{L},\ \frac{\partial K^e}{\partial x_{10}}(2,8) = -c\frac{E}{L},\ \frac{\partial K^e}{\partial x_{10}}(7,7) = c\frac{E}{L},\ \frac{\partial K^e}{\partial x_{10}}(7,8)$$

$$= c\frac{E}{L},\ \frac{\partial K^e}{\partial x_{10}}(8,8) = c\frac{E}{L} \tag{A2.5}$$

A2.3 POLYNOMIAL IN COMPUTING TIME

Computing (running) time of an algorithm is generally referred to by the time complexity and is expressed as the amount of time taken by the algorithm for some size n of the input to the problem. It denotes the total number of elementary operations required by the algorithm to solve the problem of size n. Time complexities are classified as constant, linear, logarithmic, polynomial, exponential, etc. Of these, the polynomial and exponential are the most prominent ones. An algorithm A is said to be a polynomial-time algorithm for a problem P if the number of operations required to solve P by applying A is given by a polynomial on the size of the input, or bounded by a polynomial function $f(n) \leq \tau n^k,\ \tau > 0$. It is usually expressed as $f(n) = O(n^k)$ using the big 'O' notation (Section 1.7, Chapter 1). For example, $O(n)$ is a linear time algorithm and $O(n^2)$ a quadratic time algorithm. On the other hand, algorithms with exponential running times are not polynomial, i.e., if the computing time is upper bounded by $2^{poly(n)}$, where $poly(n)$ is some polynomial in n and formally expressed as by $f(n) = O\ (2^{nk})$ for some constant k. The familiar operations such as addition, subtraction, multiplication, and division, including square roots, powers, and logarithms are considered to be performed in polynomial time. Exponential-time algorithms are obviously inefficient, since the execution time grows fast as the problem size n increases. An example for the exponential-time algorithm is the brute-force technique in solving the travelling salesman problem (Table 1.3, Chapter 1). Problems that can be solved by a polynomial-time algorithm are also called tractable problems. Khachian (1979) used the first polynomial-time algorithm for linear programming by simplex method. Karmarkar (1984) presented an efficient polynomial time algorithm for the simplex method with $f(n) = O(n^{3.5}L^2)$ as compared to $f(n) = O(n^6L^2)$ of the method used by Khachian. n is the dimension of the problem and L is the number of bits in the input.

A2.4 SYSTEM RELIABILITY AND RELIABILITY INDEX

Here we provide a brief account of the method of computing the reliability index for a system involving a set of random parameters and a random environment (e.g. external loading). We refer to the following references (Ang and Tang 1984, Melchers 1999, Maymon 1998, Nowak and Collins 2000, Manohar and Gupta 2003) for a more comprehensive treatment of the subject.

Suppose that $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)^T$ represents the system states including the loading effects. If $g(\boldsymbol{X}) := g(X_1, X_2, \ldots, X_n)$ is taken as the performance function,

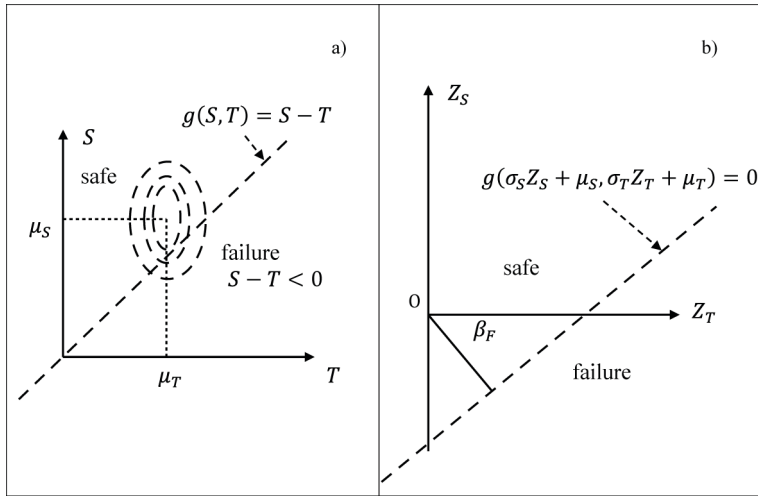


FIGURE A2.1 System reliability in a two-dimensional case; (a) failure surface in S and T (normal random variables) and (b) failure surface in reduced variates, Z_S and Z_T - standard normal variables.

$g(\boldsymbol{X}) = 0$ defines the limit state function representing the failure surface (see Figures 2.28, Chapter 2, and also Figure A2.1) which is, in general, nonlinear in $\boldsymbol{X}$. The probability of failure is then defined as:

$$P(\text{failure}) = \int_{g(X)<0} f(\boldsymbol{X})\, d\boldsymbol{X} \tag{A2.6}$$

$f(x)$ is the joint *pdf* of the state vector $\boldsymbol{X}$. The above integral is multi-dimensional and is difficult to evaluate – analytically or otherwise. For example, in the two-dimensional linear case (Figure A2.1), letting the loading effect $T = X_1$ and the system mechanical impedance (strength) $S = X_2$, the performance function is given by:

$$g(X_1, X_2) = g(S,T) = S - T \tag{A2.7}$$

Suppose that the probability distributions of S and T are known with respective means μ_S and μ_T and variances σ_S^2 and σ_T^2. In case S and T are (uncorrelated) normals, $M := S - T$ is also normal. Now, $M < 0$ defines system failure (see Figure A2.2) whose probability is given by:

$$\mathrm{P}(\text{failure}) = \int_{M<0} f_M(m)\, dm = P(M < 0) = F_M(0)$$

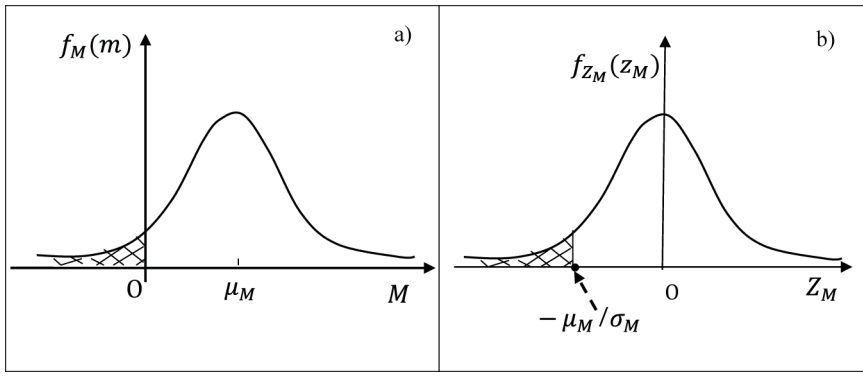


FIGURE A2.2 (a) *pdf* of limit state function $M = S - T$ and probability of failure $P(M < 0)$ shown by the hatched area and (b) *pdf* of limit state function in terms of reduced variate $Z_M = \dfrac{M - \mu_M}{\sigma_M}$ and probability of failure $P\left(Z_M < -\dfrac{\mu_M}{\sigma_M}\right)$ shown by the hatched area.

$$\Rightarrow F_M(0) = P\left(\frac{M - \mu_M}{\sigma_M} + \frac{\mu_M}{\sigma_M} < 0\right)$$

$$\Rightarrow P\left(\frac{M - \mu_M}{\sigma_M} < -\frac{\mu_M}{\sigma_M}\right) = P\left(Z_M < -\frac{\mu_M}{\sigma_M}\right), Z_M - \text{standard normal}$$

$$= \Phi\left(-\frac{\mu_M}{\sigma_M}\right) \tag{A2.8}$$

Here Φ (.) is the probability distribution function of a standard normal variable. Thus, if $\beta_F := \dfrac{\mu_M}{\sigma_M}$, the probability of failure is given by $\Phi(-\beta_F)$ and β_F is known as the reliability index. In terms of reduced variates Z_S and Z_T defined as $Z_S = \dfrac{S - \mu_S}{\sigma_S}$ and $Z_T = \dfrac{T - \mu_T}{\sigma_T}$, the first two moments of M are given by $\mu_M = \mu_S - \mu_T$ and $\sigma_M^2 = \sigma_S^2 + \sigma_T^2$. Therefore, the probability failure is given by:

$$\Phi\left(-\frac{\mu_M}{\sigma_M}\right) = \Phi\left(-\frac{\mu_S - \mu_T}{\sqrt{\sigma_S^2 + \sigma_T^2}}\right) \tag{A2.9}$$

Thus $\beta_F = \dfrac{\mu_S - \mu_T}{\sqrt{\sigma_S^2 + \sigma_T^2}}$. Now, the limit state function $S - T = 0 \Rightarrow Z_s\sigma_s - Z_T\sigma_T + \mu_S - \mu_T = 0$ is a straight line (Figure A2.1) in the space of reduced variates. It may be observed that the perpendicular distance from the origin to the straight line is $\dfrac{\mu_S - \mu_T}{\sqrt{\sigma_S^2 + \sigma_T^2}}$ which leads to the result that the reliability index β_F is the minimum distance from the origin of the standard normal space to the surface of the limit state function g.

A generalization of the above result is as follows.

If $g(\boldsymbol{X})$ is a function of uncorrelated normal variables, the reliability index β_F is the least distance from the origin of the space of reduced variates to the hypersurface $g\left(\sigma_{X_1} Z_1 + \mu_{X_1}, \sigma_{X_2} Z_2 + \mu_{X_2}, \ldots, \sigma_{X_n} Z_n + \mu_{X_n}\right) = 0$, where $Z_i = \dfrac{X_i - \mu_{X_i}}{\sigma_{X_i}}$, $i = 1, 2, \ldots, n$. This implies that if Z_i^*, $i = 1, 2, .., n$ is the point on the hypersurface nearest from origin, $\beta_F = \left(\sum_{i=1}^{n} \left(Z_i^*\right)^2\right)^{1/2}$. The point $\left(Z_1^*, Z_2^*, ..Z_n^*\right)$ is known as the most probable failure point or the design point.

The probability of failure is thus obtainable through computation of β_F instead of evaluation of the multi-dimensional integral $\int_{M<0} f_M(m)\,dm$ in Equation (A2.8). If $\boldsymbol{X}$ is normal (uncorrelated), Z_i^*, $i = 1, 2, \ldots$ may be obtained by constrained minimization of the function $\sum_{i=1}^{n} \left(Z_i\right)^2$ under the constraint $g\left(\sigma_{X_1} Z_1 + \mu_{X_1}, \sigma_{X_2} Z_2 + \mu_{X_2}, \ldots, \sigma_{X_n} Z_n + \mu_{X_n}\right) = 0$.

REFERENCES

Ang, A. H. S. and W. H. Tang. 1984. *Probability Concepts in Engineering Planning and Design, Volume II Decision, Risk and Reliability* (*Vol. II*), p. 194. New York: John Wiley and Sons Ltd.

Karmarkar, N. 1984. A new polynomial time algorithm for linear programming. *Combinatorica* 4: 373–395.

Khachian, H. G. 1979. A new polynomial-time algorithm for linear programming. [In Russian.] *Koklady Adademii Nauk SSSR* 244: 1093–1096.

Manohar, C. S. and S. Gupta. 2003. Modelling and evaluation of structural reliability: Current status and future directions. In K. S. Jagadish, R. N. Iyengar (eds.), *Research Reviews in Structural Engineering*, Golden Jubilee Publications of Department of Civil Engineering, IISc., Bangalore (University Press).

Maymon, G. 1998. *Some Engineering Applications in Random Vibrations and Random Structures.* AIAA. Inc.

Melchers, R. E. 1999. *Structural Reliability, Analysis and Prediction.* 2nd Ed. John Wiley and Sons Ltd.

Nowak, A. S. 2000. *Reliability of Structures*. New York: McGraw-Hill Companies Inc.

Appendix 3

A.3.1 MONTE CARLO (MC) SIMULATION OF RANDOM VARIABLES (RVS) WITH SPECIFIED PROBABILITY DISTRIBUTION

MC simulation aims at sampling of RVs with specified probability distributions. For simple probability distributions, sampling may be easy by methods such as inversion, rejection and importance sampling. These methods use transformation of RVs along with random number generation (see Appendix 1). We consider a few examples to highlight these straightforward techniques of inversion, rejection and importance sampling.

A3.1.1 Inversion Method of Sampling RVs

Random numbers with a specified probability distribution (other than uniform) are obtained via an appropriate transformation. Considering two scalar random variables X and Y, we can write:

$$F_Y(y) = F_X(x) \Rightarrow y = F_Y^{-1}\left[F_X(x)\right] \tag{A3.1}$$

In order to make sense of the last identity involving the inverse of F_Y, it is assumed that F_X and F_Y are absolutely continuous* with respect to each other.

---

* *Absolutely continuous functions*
Absolute continuity of a function is a smoothness property and is stronger than uniform continuity. A function $f(x)$ is absolutely continuous on an interval $[a,b] \in \mathbb{R}$ if for each $\varepsilon > 0$, there exists a $\delta > 0$ such that $\sum_{i=1}^{n} \left| f(b_i) - f(a_i) \right| < \varepsilon$ for a finite collection of non-overlapping sub-intervals $\left\{ [a_i, b_i], 1 \le i \le n \right\}$ in the interval $[a,b]$, satisfying the condition $\sum_{i=1}^{n} (b_i - a_i) < \delta$. If two functions $f(x)$ and $g(x)$ are absolutely continuous having the same support, they are absolutely continuous to each other.

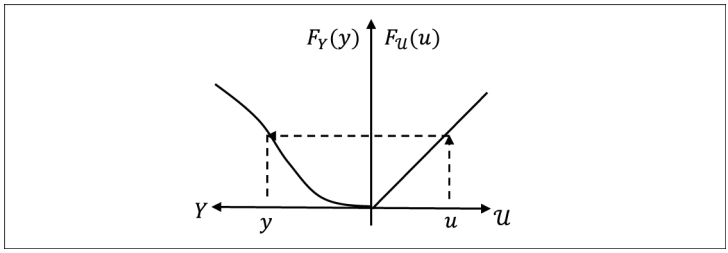


FIGURE A3.1 Generation of realizations for X of a specified $F_Y(y)$ via a transformation using uniformly distributed random variable.

Today's computing machines are equipped with random number generators that provide random numbers uniformly distributed in $[0,1]$. Thus, with a set of numbers $u_1, u_2, \ldots$. (generated as realizations of the random variable $\mathcal{U} \sim U(0,1)$), the corresponding set for the random variable Y is obtained via the inverse transformation:

$$y_i = F_Y^{-1}(u_i),\ i = 1,2\ldots \tag{A3.2}$$

The numerical generation of a random variable Y is illustrated graphically in Figure A3.1.

A3.1.2 Simulation of a Discrete RV by Inversion Method

In the following example, we use the inversion method to realize samples of a discrete probability distribution.

Example A3.1. Consider a discrete RV $X(\Omega) \to S$ where $\Omega = \{\omega_1, \omega_2, \omega_3\}$ and $S = \{1,2,3\}$. The given discrete probabilities are $p_1 = P(\omega_1 = 1) = 0.287$, $p_2 = P(\omega_2 = 2) = 0.467$ and $p_3 = P(\omega_2 = 3) = 0.246$. It is required to sample the distribution.

Solution. For instance, the three outcomes ω_1, ω_2 and ω_3 may denote the possible three weather conditions on a day, i.e., $\Omega = (\text{cloudy}, \text{rainy}, \text{sunny})$. The *pdf* and CDF are shown in Figure A3.2. To simulate Y, we use the transformation (Figure A3.1):

$$P(Y \le y) = F_Y(y) = F_{\mathcal{U}}(u) = P(\mathcal{U} \le u) \tag{A3.3}$$

with $\mathcal{U} \sim U(0,1)$, i.e., $\mathcal{U}$ is a uniformly distributed RV in the interval $[0,1]$. Note that, for a given u, it is obvious from Figure A3.1 that $F_{\mathcal{U}}(u) = u$. The realization y is hence obtained by the inversion:

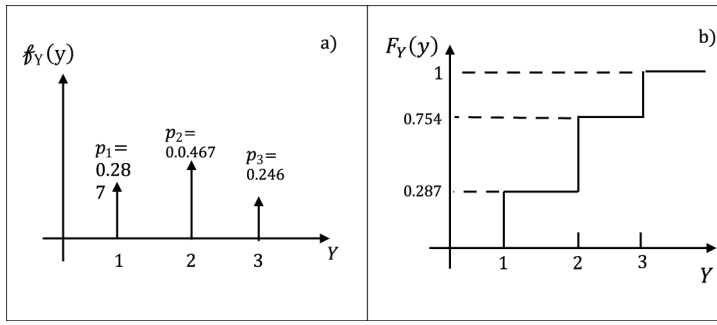


FIGURE A3.2 (a) *pdf* and (b) CDF of the discrete random variable *Y*.

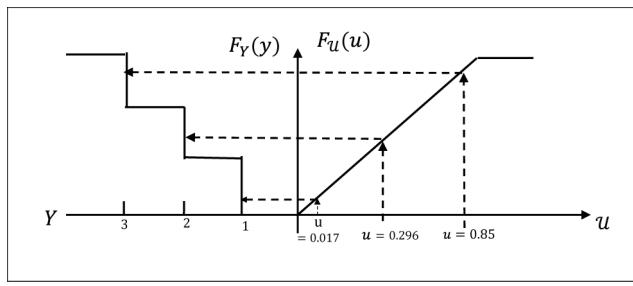


FIGURE A3.3 Sampling of the discrete RV*Y*; $F_Y(y)$– CDF of the discrete RV*Y*, $F_U(u)$ – CDF of uniformly distributed RV; arrows marked '1', '2' and '3' indicate the realizations of the RV *Y*. Note that the figure is not drawn to scale.

$$y = F_Y^{-1}(u) \tag{A3.4}$$

For a discrete RV, no explicit inversion is necessary. Figure A3.3 shows the two distributions together, which is the discrete version of Figure A3.1. To realize *y*, we will first have a u from the random number generator. If $F_U(u) = u \le p_1 = 0.2874$, then we take $y = 1$ (cloudy). If $u \le p_1 + p_2 = 0.7542$, then we identify $y = 2(\text{rainy})$, Lastly, If $u \le p_1 + p_2 + p_3 = 1.0$ then $y = 3(\text{sunny})$.

While performing the simulation to get samples in Figure A3.3, let u be obtained as 0.296. With this value of $u > p_1$ and $\le p_1 + p_2 = 0.7542$, we realize $y = 2$ (as marked in the figure) and identify the weather condition on the particular day as 'rainy'. Two more states are identified and marked in the figure for other typical observations of *u*.

A3.1.3 Simulation of a Continuous RV by Inversion Method

The inverse transformation method is advantageous when $F_Y^{-1}(x)$ is known explicitly in terms of *x*.

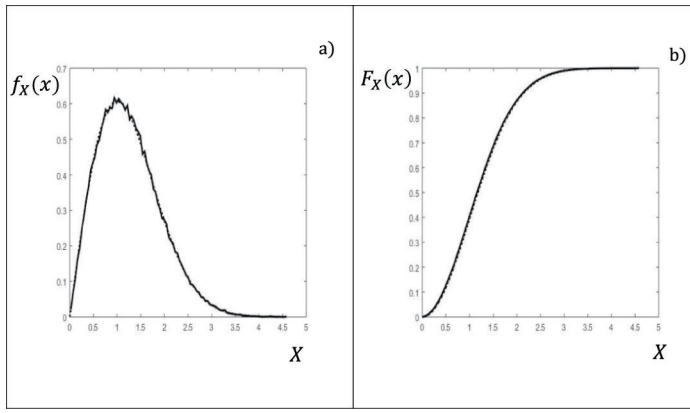


FIGURE A3.4 Sampling by inversion method of Rayleigh RV X: (a) simulated *pdf* and (b) simulated CDF, theoretical *pdf* and CDF shown in dashed lines.

Example A3.2. Suppose that we are required to simulate the Rayleigh RV X with *pdf*:

$$f_X(x) = x \exp\left(-\frac{x^2}{2}\right), 0 \le x < \infty. \tag{A3.5a}$$

Solution. In this case, the CDF of the Rayleigh RV X is:

$$F_X(x) = \int_0^x x \exp\left(-\frac{x^2}{2}\right) dx = 1 - \exp\left(-\frac{x^2}{2}\right), \ 0 \le x < \infty. \tag{A3.5b}$$

As per Equation (A3.4), one has:

$$x_i = F_X^{-1}(u_i), \quad i = 1, 2 \ldots$$

$$= \sqrt{\left(-2\log\left(1 - u_i\right)\right)} \tag{A3.6}$$

Figure A3.4 shows the *pdf* and CDF of the simulated Rayleigh RV. The theoretical graphs are also indicated in the figure.

Obviously, the method is feasible if the inversion (in Equation A3.2) is possible. Most practical requirements may involve complicated expressions for probability distributions rendering the inversion method difficult to adopt. In addition, one may face difficulty when it is required to sample a RV X with *pdf* $f_X(x)$ explicitly unknown. For instance, $f_X(x)$ may be known only up to a normalizing constant. Simulating the Boltzmann distribution in simulated annealing method of optimization (Section 3.4.2, Chapter 3) is one striking example. Methods like rejection sampling and importance sampling are some of the popular alternatives.

A3.1.4 Rejection Sampling (von Neumann 1951)

Suppose that $f_X(x)$ is the target *pdf*. The rejection method uses a proposal *pdf* $h_X(x)$ which is easier to simulate than $f_X(x)$. We select $h_X(x)$ such that:

$$f_X(x) \leq Mh_X(x), \forall x \qquad M \in \mathbb{R} \tag{A3.7}$$

The method involves the following steps to obtain the realizations for the target density $f_X(x)$.

(i) Generate random sequence $\{x_i\}$ according to $h_X(x)$ (since realizations from $h_X(x)$ can be more readily generated using Equation A3.4).

(ii) Generate a random sequence $\{u_i\}$ uniformly distributed over $\left(0, Mh_X(x_i)\right)$. The point (x_i, u_i) will be from a uniform distribution in the area below $Mh_X(x)$.

(iii) If $u_i \leq f_X(x_i)$, then accept x_i as a realization from $f_X(x)$. The point (x_i, u_i) is positioned in the area below the $f_X(x)$ curve (e.g. (x_2, u_2) in Figure A3.5

(iv) If $u_i > f_X(x_i)$, then reject x_i (e.g. (x_1, u_1) in Figure A3.5) and return to step i).

The point (x_1, u_1) in Figure A3.5 is rejected and (x_2, u_2) accepted, i.e. x_2 is an acceptable realization from $f_X(x)$.

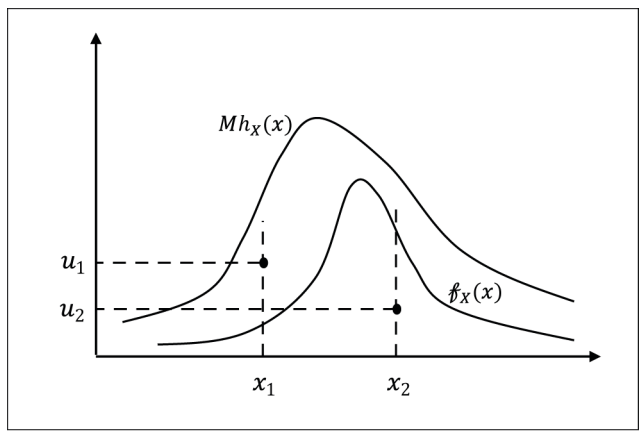


FIGURE A3.5 Sampling by rejection method of the target density $f_X(x)$; x_1; x_1 is to be rejected since $u_1 > f_X(x_1)$ and x_2 is an acceptable realization from $f_X(x)$ since $f_X(x_2)$.

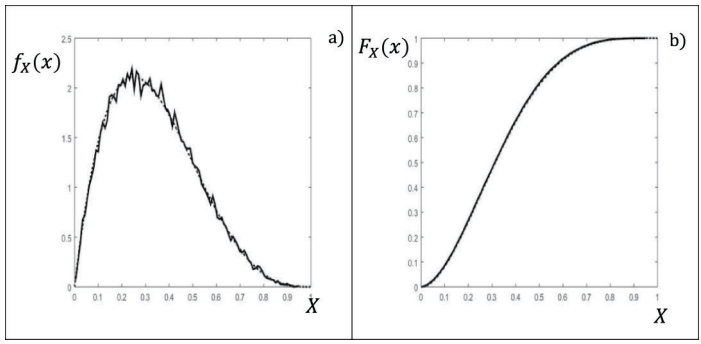


FIGURE A3.6 Sampling from beta distribution by rejection method: (a) simulated *pdf* and (b) simulated CDF; theoretical *pdf* and CDF are also shown in dashed lines.

Further details on the method and validation of the above procedural steps may be found in Roy and Rao (2017). The application of the method is illustrated via the following example.

Example A3.3. Consider sampling by the rejection method of the Beta distribution with *pdf* $f_X(x) = 20x(1-x)^3$, $0 < x < 1$.

Solution. Often, it may be difficult to simulate the target *pdf* by the inversion method. To use the rejection method, we take the proposal *pdf* $h_X(x)$ to be uniform over $(0,1)$. To find the constant M such that Equation (A3.7) is satisfied, we maximize $\frac{f_X(x)}{h_X(x)} = 20x(1-x)^3$. This leads to $M = \frac{135}{64}$. Figure A3.6 shows the *pdf* and CDF of the generated random sequence for the beta distribution. Comparisons with the theoretical *pdf* and CDF are also shown in the figure.

A3.1.5 Importance Sampling Method (Rubinstein 1981)

In addition to the fundamental problem of obtaining samples from a probability distribution, one often wishes to evaluate expectations such as:

$$E[g(x)] = \int_{\mathbb{R}} g(x) f_X(x) dx \tag{A3.8}$$

where $g(x) : \mathbb{R} \to \mathbb{R}$. See Appendix 1 for details on the expectation operator $E[.]$. Let I denote the integral in the last equation. When f_X has a standard form, e.g. Rayleigh or exponential, it is straightforward to sample from it by using the inversion method. If the sampled values are $x_1, x_2 \ldots, x_N$, then one has the following sampled average as an approximation to the integral I:

$$I_N(g) = \frac{1}{N} \sum_{i=1}^{N} g(x_i) \tag{A3.9}$$

To denote that $x_i, i = 1, 2, \ldots, N$ are sampled from the *pdf* $f_X(x)$, we usually write $x_i \sim f_X(x)$. As $N \to \infty$ there is a finite probability that $I_N(g)$ does not deviate much from $E[g(x)]$.

In fact, by the law of large numbers,† one may have:

$$\lim_{N\to\infty} I_N(g) = E[g(x)] \tag{A3.10}$$

In many instances, direct sampling from the target density $f_X(x)$ is difficult. Importance sampling technique is a method useful for sampling particularly from high-dimensional, complicated distributions. As in the rejection method, we use a proposal *pdf* $p_X(x)$ whose support‡ includes that of $f_X(x)$. Write the integral I as:

$$I = \int_{\mathbb{R}} g(x) f_X(x) dx = \int_{\mathbb{R}} \left(g(x) \frac{f_X(x)}{p_X(x)} \right) p_X(x) dx = E_p \left[\left(g(x) \frac{f_X(x)}{p_X(x)} \right) \right] \tag{A3.11}$$

The subscript 'p' in $E_p[.]$ indicates that the expectation is with respect to the *pdf* $p_X(x)$. We have an estimate for the original integral I as:

$$\int_{\mathbb{R}} g(x) f_X(x) dx \cong \frac{1}{N} \sum_{i=1}^{N} \frac{f_X(x_i) g(x_i)}{p_X(x_i)} \tag{A3.12}$$

In Equation (A3.12), $x_i \sim p_X(x)$. We express the sampled average in the last equation by $I_N\left(\frac{gf}{p}\right)$. It is expected that with a finite N, the RV $I_N\left(\frac{gf}{p}\right)$ might have smaller variance. In this context, the sample variance is:

$$\text{var}\left(I_N\left(\frac{gf}{p}\right) \right) = E_p\left[\left(\frac{g(x) f_X(x)}{p_X(x)} \right)^2 \right] - I^2 \tag{A3.13}$$

---

† *Law of large numbers (LLN)*
The law of large numbers is associated with the asymptotic nature of probabilities of events, empirically estimated, for instance, from actual observations/experiments. It is concerned with the convergence of these estimates. There exist two versions of the law of large numbers, e.g. the weak and strong laws. Consider $X_j, j = 1,2,..,N$ to be a sequence of independent random variables, each having a finite common mean, $E[X_j] = m, j = 1,2,\ldots,N$ and let $x_1, x_2, \ldots, x_N$ be the samples drawn for these RVs. The weak law of large numbers says that for a specified large number of trials of a random experiment, the sample mean $\frac{x_1 + x_2 + \ldots + x_N}{N}$ stays near the value m and may differ, though infrequently, by large values. On the other hand, the strong law of large numbers states that, with probability 1, the difference of the sample mean from m may be made smaller than any given positive ε for sufficiently large *N*. See Roy and Rao (2017) for proofs.

‡ *Support of a function* (Rudin 1976)
Support of a function $f(x)$ is the set of points contained in the domain of f where the function is non-zero. Similar definition applies in probability theory where the support of a RV X is defined as the set $S = \{x \in \mathbb{R}: f_X(x) > 0\}$

The above expression is in line with the definition of variance of a RV X:

$$\text{var}(X) = E\left[\left(X - E[X]\right)^2\right] = E[X^2] - \left(E[X]\right)^2 \tag{A3.14}$$

The importance sampling method principally focuses on providing guidelines on the selection of the sampling distribution $p_X(x)$ to obtain accurate estimates with reduced sample variance. It can be shown (Dimov 2008) that if $p_X(x) \propto |f_X(x)g(x)|$, i.e. $p_X(x) = c|f_X(x)g(x)|$ with c being a constant, the estimate will have the minimum variance. Also see the discussion in Roy and Rao (2017) on the choice of the sampling *pdf* $p_X(x)$. Specifically for an integral as in Equation (A3.11), the main consideration in the choice of $p_X(x)$ is to have $\frac{f_X(x)}{p_X(x)} < 1$ whenever $g(x)f_X(x)$ in the integral is large, i.e. to have $f_X(x)$ to be dominated by $p_X(x)$ in the sub-domain(s) of $\mathbb{R}$ where $(x)f_X(x)$ has larger absolute values.

In regions of relatively smaller $g(x)f_X$, one may have $\frac{f_X(x)}{p_X(x)} > 1$; but this condition is of less relevance since contributions to the integral from these regions are insubstantial. The following example clarifies some of these issues related to the implementation of the method.

Example A3.4. We evaluate the integral $I = \int_{\mathbb{R}} x^4 e^{x^2/4} I_{\{x\le 2\}} \cdot \left(\frac{e^{-x^2/2}}{\sqrt{2\pi}}\right) dx$ by the importance sampling method. $I_{\{x\le 2\}}$ is an indicator function defined by:

$$\begin{aligned} I_{\{x\le 2\}} &= 1, \text{ for } x \le 2 \\ &= 0, \text{ otherwise} \end{aligned} \tag{A3.15}$$

Solution. One way of evaluating I is by treating the integrand as $g(x)f(x)$ and posing the integral as $E_f\left[g(x)\right]$ where $g(x) = x^4 e^{x^2/4} I_{\{x\le 2\}}$ and the expectation operator is with respect to the *pdf* $f_X(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$. Without resorting to the importance sampling method, we sample directly from $f_X(x)$ which is recognized as the normal *pdf* corresponding to $X \sim \mathcal{N}(0,1)$. Using these samples, the MC estimate I_N for I is obtained from 10 independent MC runs and with $N = 2E4$ and is shown (in plus sign) in Figure A3.7.

Now, introducing a new sampling *pdf* $h(x)$ which is chosen to be a normal with non-zero mean 2, the integral I may be written as:

$$I = \int_{\mathbb{R}} \frac{g(x)f(x)}{h(x)} h(x)dx = \int_{\mathbb{R}} \frac{\left(x^4 \; e^{x^2/4} \; I_{\{X\le 2\}}\right)\left(\frac{e^{-x^2/2}}{\sqrt{2\pi}}\right)}{\left(\frac{e^{-(x-\mu)^2/2}}{\sqrt{2\pi}}\right)} \left(\frac{e^{-(x-\mu)^2/2}}{\sqrt{2\pi}}\right) dx$$

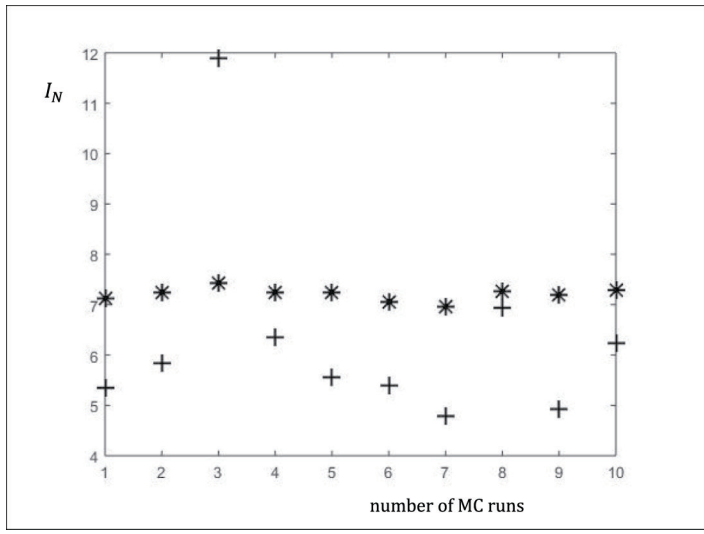


FIGURE A3.7 MC estimate I_N of the integral I of Example A3.4 with N = 20,000: (a) estimate without important sampling in + sign and (b) estimate with important sampling in * sign.

$$= E_h\left[\frac{\left(x^4\ e^{x^2/4}\ I_{\{X\le 2\}}\right)\left(\frac{e^{-x^2/2}}{\sqrt{2\pi}}\right)}{\left(\frac{e^{-(x-\mu)^2/2}}{\sqrt{2\pi}}\right)}\right] \tag{A3.16}$$

$E_h(.)$ in Equation (A3.16) stands for expectation with respect to the new sampling *pdf* $h(x)$. That is, an estimate for I is obtained by sampling $X \sim h(x) = \mathcal{N}(2,1)$ and averaging similar to Equation A3.12. The new estimate which is better than the original one is also shown (in the star sign) in Figure A3.7. Sampling efficiency may be observed in Figure A3.8 from a comparison of standard deviations σ of the two estimates.

With a proper choice of μ, one may improve the sampling efficiency and have an improved variance reduction in obtaining the estimate (see Roy and Rao 2017).

A3.2 MARKOV CHAINS (STROOK 2005, NORRIS 2012)

A Markov chain is a discrete stochastic process. A stochastic process is a parametrized family of random variables $X = \{X_t(\omega): t \ge 0, t \in T, \omega \in \Omega\}$ where t is often a scalar (real) parameter defined on an index set T (finite or countable or even uncountable). $T \in \mathbb{R}^+$ may be the familiar time axis (say, an interval on the time axis). As defined in Appendix 1, a random variable (RV) denotes a mapping of the probability space Ω

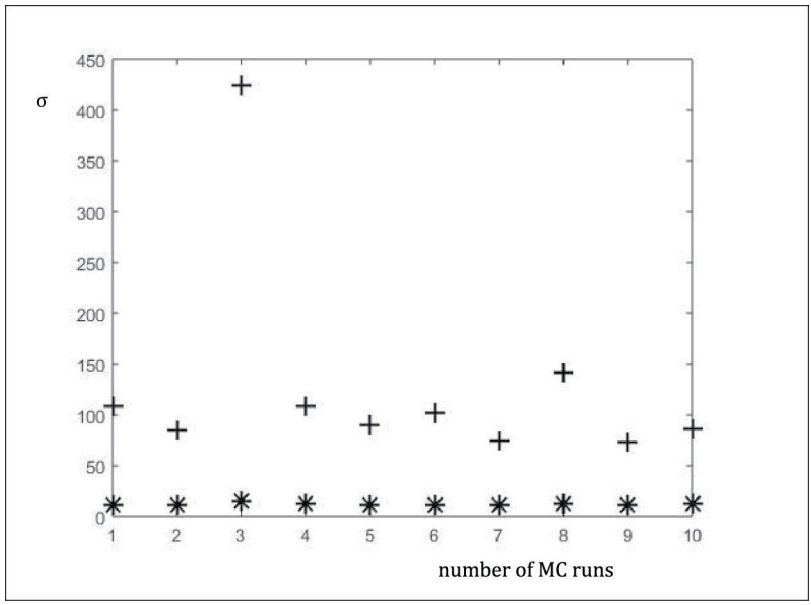


FIGURE A3.8 MC estimate I_N of the integral I of Example A3.4 with N = 20,000, standard deviation σ of the estimate: (a) without important sampling in + sign and (b) with important sampling in * sign.

to $\mathbb{R}$ with $\boldsymbol{\omega}$ representing the possible outcomes of the RV and constituting Ω. For a fixed $\omega^* \in \Omega$, $X_t(\omega^*)$ is often called a path (or realization or trajectory) of the stochastic process. On the other hand, for a fixed $t^* \in T$, $X_{t^*}(\omega)$ is an RV. In Appendix 4, a detailed description of stochastic processes is provided which is fundamental to the development of many stochastic optimization methods.

In contrast to the continuous time and continuous state stochastic process defined above, a discrete Markov chain may be visualized as an ensemble or collection of RVs $\{X_0, X_1, \ldots\}$ evolving at discrete time instants. Moreover, the RVs may map Ω to a finite state space $S = \{a_0, a_1, \ldots, a_m\}$ with each $a_m \in \mathbb{R}$. Markov chains were first introduced by the Russian mathematician Andrei Andreyevich Markov (1856–1922). In a Markov chain, the sequence of random variables exhibits the Markov property: the present state depending only on the immediate past. Expressed in terms of conditional probabilities, the property is expressible as:

$$P\left(X_{n+1} = a_j | X_n = a_i, X_{n-1} = a_{i-1}, \ldots, X_0 = a_0\right) = P(X_{n+1} = a_j | X_n = a_i) \tag{A3.17}$$

This is analogous to (or the discrete analogue of) the definition of a Markov process (Roy and Rao 2017). A stochastic process is called Markov if, conditioned on the current

state, its future is independent of its past. In Equation (A3.17), $P(X_{n+1} = a_j \mid X_n = a_i)$ is known as the transition probability (transition kernel) denoted by $\mathcal{T}_{ij}$ – the probability that X_{n+1} assumes the value a_j given that $X_n = a_i$ at the previous step (must be understood in the sense of a density in the continuous case). For a finite state Markov chain with state space $S = \{a_0, a_1, \ldots, a_m\}$, we define a transition probability matrix as:

$$\boldsymbol{\mathcal{T}} = \left[\mathcal{T}_{ij}\right] \in \mathbb{R}^{m+1 \times m+1}, 0 \le i, j \le m \tag{A3.18}$$

Here $0 \le \mathcal{T}_{ij} \le 1$ and $\sum_j \mathcal{T}_{ij} = 1 \forall i$. (i.e., each row is a distribution over *S*). It follows that the ij^{th} entry of the matrix $\mathcal{T}^n$ (n^{th} power of $\boldsymbol{\mathcal{T}}$) gives the probability that the Markov chain, starting in state a_i, will be in state a_j after *n* steps.

The matrix $\boldsymbol{\mathcal{T}}$ is useful to know the probability of the present state (after, say, *n* transitions), given the corresponding probability of the initial state $X_0 = a_i$, $i = 1, 2, \ldots, m$. Thus if $\boldsymbol{p}^{(0)} = \left(p_1^{(0)}, p_2^{(0)}, \ldots, p_m^{(0)}\right)$ is the vector of initial probabilities, then the probability of the states after *n* transitions is obtained as:

$$\boldsymbol{p}^{(n)} = \boldsymbol{p}^{(0)} \boldsymbol{\mathcal{T}}^n \tag{A3.19}$$

This is the same as $\boldsymbol{p}^{(n)} = \left(\boldsymbol{\mathcal{T}}^n\right)^T \boldsymbol{p}^{(0)}$ except that $\boldsymbol{p}^{(0)}$ is a row vector in the former case and a column vector in the latter. Equation (A3.19) is known as the Kolmogorov-Chapman equation. It may often be the case that all the elements of $p^{(0)}$ are zero except for a single entry of unity corresponding to the state where the process starts. The transition matrix $\boldsymbol{\mathcal{T}}$ is regular if any power of the matrix contains all positive (non-zero) entries. A Markov chain is regular if its $\boldsymbol{\mathcal{T}}$ is.

Example A3.5. Consider RVs $\{\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots\}$ representing the weather conditions – cloudy, rainy and sunny – on consecutive days with each RV $\boldsymbol{X}_n : \Omega \to S$. Here $\Omega = \{\text{cloudy, rainy and sunny}\}$ is the probability space and $S = \{1, 2, 3\} = \boldsymbol{X}_n(\Omega)$ on $\mathbb{R}$. Let us take the transition probability matrix $\boldsymbol{\mathcal{T}}$ as:

$$\boldsymbol{\mathcal{T}} = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.4 & 0.5 \end{bmatrix} \tag{A3.20}$$

Solution. It is easy to verify that all powers of $\boldsymbol{\mathcal{T}}$ have positive (non-zero) entries and the Markov chain is regular. Suppose that $p^{(0)} = \{0, 1, 0\}$. The distribution of the states at the end of, say, five steps is

$$p^{(5)} = p^{(0)} \boldsymbol{\mathcal{T}}^5 = \begin{bmatrix} 0.29 & 0.465 & 0.245 \end{bmatrix} \tag{A3.21}$$

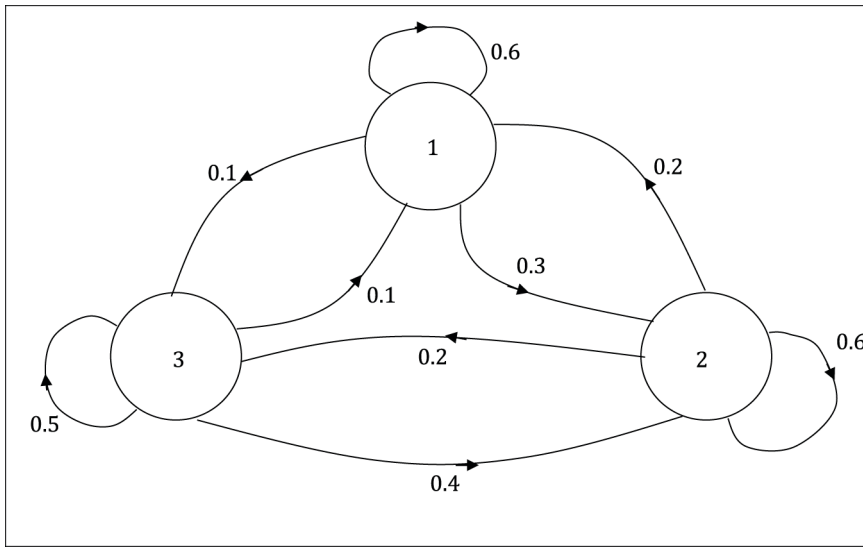


FIGURE A3.9 Directed graph for the transition probability matrix $\mathcal{T}$ in Example A3.5; state space $S=(1,2,3)$ on $\mathbb{R}$ and the probability space $\Omega = X_n^{-1}(S) = (\text{cloudy}, \text{rainy}, \text{sunny})$ where the RV $X_n : \Omega \to S$.

The interpretation of the result is that while it is a fully rainy day today, the fifth day from now may be cloudy with probability = 0.29, rainy with probability = 0.465 and sunny with probability = 0.245.

The transition probability matrix $\mathcal{T}$ can be described by a directed graph whose vertices are the states and edges with arrows are the transition paths from a state a_i to a_j with probability $\mathcal{T}_{ij}$. For the last example, Figure A3.9 shows the directed graph for $\mathcal{T}$ in Equation (A3.20).

A3.2.1 Irreducibility of a Markov Chain

If all states in a Markov chain communicate with each other, it is irreducible. Two states a_i and a_j communicate with each other – denoted by $a_i \leftrightarrow a_j$ – if for all $n_1, n_2 \geq 0$ one has:

$$\mathcal{T}_{i,j}^{n_1} > 0 \text{ and } \mathcal{T}_{j,i}^{n_2} > 0 \tag{A3.22}$$

If a Markov chain is irreducible, every state a_j is eventually reachable, starting from any other state a_i, i.e: $\mathrm{P}\left(\boldsymbol{X}_n = a_j | \boldsymbol{X}_0 = a_i\right) > 0$, for some $n \geq 0$. Irreducibility is an equivalence relation in that $a_i \leftrightarrow a_j$ and $a_j \leftrightarrow a_k$ is equivalent to $a_i \leftrightarrow a_k$. The Markov chain of Example A3.5 is irreducible.

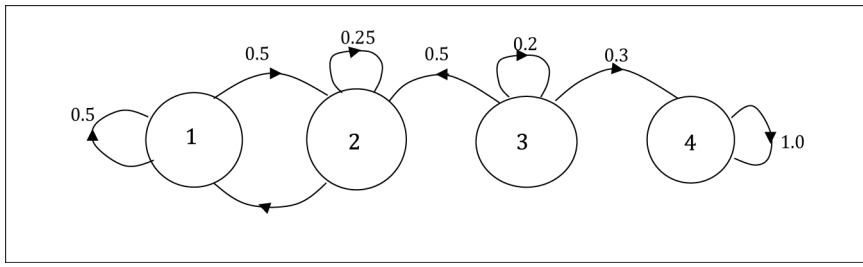


FIGURE A3.10 Transition graph for Markov chain with the matrix $\mathcal{T}$ given in Equation (A3.23).

Suppose that we have a four-state Markov chain with the transition matrix:

$$\mathcal{T} = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0.75 & 0.25 & 0 & 0 \\ 0 & 0.5 & 0.2 & 0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A3.23}$$

The corresponding transition graph is shown in Figure A3.10. The Markov chain is reducible in that the state space S may be reduced to non-overlapping sets as $\{1,2\}$, $\{3\}$ and $\{4\}$.

A3.2.2 Periodicity of a Markov Chain

Periodicity of a state a_i is:

$$d_i = \gcd\left(n \geq 0, \mathcal{T}_{i,i}^n > 0\right) \tag{A3.24}$$

'gcd' is the greatest common divisor. A familiar example for a periodic Markov chain is the one with the transition matrix:

$$\mathcal{T} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{A3.25}$$

Here $S = \{a_0, a_1\}$. We find that $\mathcal{T}^{2l} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\mathcal{T}^{2l+1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ for all $l = 1,2,\ldots$ and thus $\mathcal{T}_{1,1}^n > 0$ for even $n = 2,4,6,\ldots$ with gcd = 2. Hence both a_0 and a_1 are of period 2. Therefore, the Markov chain is periodic. If $d_i = 1, \forall i$, then a Markov chain is aperiodic. An irreducible chain may be periodic or aperiodic.

The Markov chain in Example A3.5 is aperiodic in addition to being irreducible.

A3.2.3 Stationary and Limiting Distributions of a Markov Chain

The fundamental issue in the theory of Markov chains is its long-term or asymptotic behaviour. This is characterized by the existence of a limiting distribution for the Markov chain. Before discussing limiting distributions, we first consider the concept of a stationary distribution of a Markov chain. A row vector $\mathbf{þ}$ of probabilities such that $\sum_i þ_i = 1$ is stationary (invariant) distribution of a Markov chain if:

$$\mathbf{þ}\boldsymbol{\mathcal{T}} = \mathbf{þ} \tag{A3.26}$$

It can be shown (Norris 2012) that if a Markov chain is irreducible and aperiodic, it has a unique stationary distribution which is the limiting distribution. A stationary distribution $\mathbf{þ}$ is a limiting distribution if:

$$\lim_{n\to\infty}[\mathcal{T}^n]_{ij} = þ_j, \forall\, i \tag{A3.27}$$

Example A3.6. For the Markov chain with $\boldsymbol{\mathcal{T}}$ in Equation (A3.25), we can identify the stationary distribution $\mathbf{þ}$ to be $\left(\frac{1}{2}\ \frac{1}{2}\right)$. The 2-state Markov chain may have a unique stationary distribution; but we observe that $\lim_{n\to\infty} \mathcal{T}_{ij}^{n} \neq þ_j$. Thus $\mathbf{Þ}$ is not a limiting distribution, because the chain, even though irreducible, is periodic with period = 2.

Example A3.7. The Markov chain in Example A3.5 is irreducible and aperiodic. We find the limiting distribution by Equation (A3.27).

Solution. After sufficiently many transitions ($n \geq 10$), the associated transition matrix shows the property in Equation (A3.27), i.e.,

$$\lim_{n>10}\mathcal{T}^n = \begin{bmatrix} 0.2926 & 0.4634 & 0.2440 \\ 0.2926 & 0.4634 & 0.2440 \\ 0.2926 & 0.4634 & 0.2440 \end{bmatrix} \tag{A3.28}$$

The stationary (and thus the limiting) distribution is $\mathbf{þ} =$. A few samples of the Markov chain are simulated and shown in Figure A3.11. Starting with the initial probabilities (on day 1) of the three states represented by the vector $\boldsymbol{p}^{(0)} = \{0,1,0\}$, each sample of the Markov chain is obtained by Monte Carlo simulation. The probabilities $\boldsymbol{p}^{(k)}$ of the states on any k^{th} day are generated from $\boldsymbol{p}^{(k)} = \boldsymbol{p}^{(0)}\boldsymbol{\mathcal{T}}^k$. The particular state on this day is decided by these discrete probabilities and simulated specifically by the inversion method of simulation (see Section A3.1).

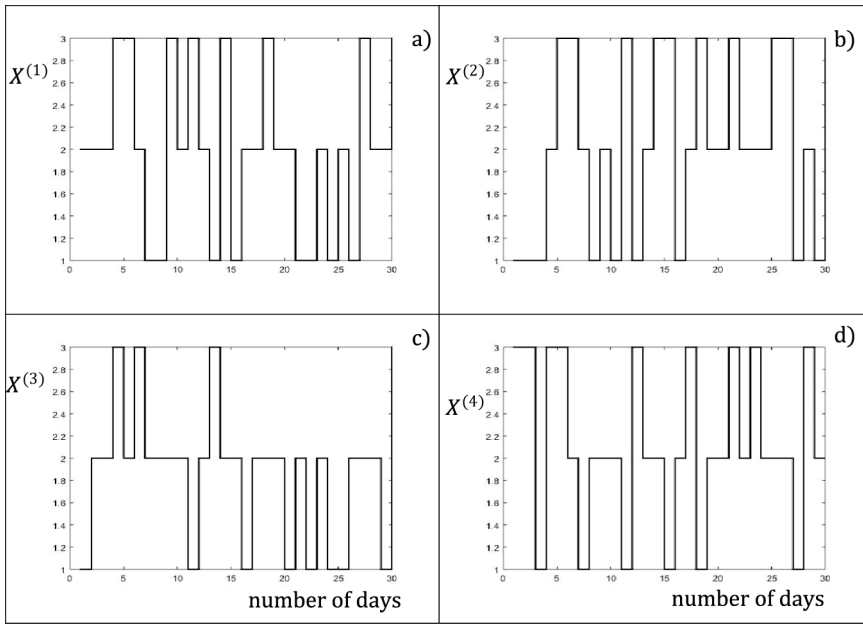


FIGURE A3.11 Typical four samples of Markov chain $X^{(k)}, k = 1,2,3,4$ – a discrete stochastic process – of Example A3.5 simulated using the transition probability matrix $\mathcal{T}$ (Equation A3.20); initial probabilities (on day 1) of the three states – $\boldsymbol{p}^{(0)} = \{0,1,0\}$; across the ensemble, states on any day denote an RV with a discrete sample space $\Omega = (1,2,3)\,\mathbb{R}$ with 1–1 correspondence to the three states 'rainy', 'cloudy' and 'sunny'.

A3.2.4 Ergodic Chains

Markov chains with the property of irreducibility and aperiodicity are known as ergodic chains. Regardless of the initial probabilities of the states, the ergodic Markov chains eventually reach the probabilities corresponding to the limiting distribution. It may be difficult, in general, to find the limiting distribution. An example is given in the following to obtain the limiting distribution utilizing the property in Equation (A3.26).

Example A3.8. For the three-state Markov chain in Example A3.5, let Þ =. The Markov chain is ergodic and we find the limiting distribution.

Solution. According to Equation (A3.26), we have $\sum_i^m Þ_i \mathcal{T}_{i,j}$, i.e.:

$$0.6Þ_1 + 0.2Þ_2 + 0.1Þ_3 = Þ_1$$

$$0.3Þ_1 + 0.6Þ_2 + 0.4Þ_3 = Þ_2$$

$$0.1Þ_1 + 0.2Þ_2 + 0.5Þ_3 = Þ_3 \qquad \text{(A3.29)}$$

The above equations are linearly dependent.** Combined with the equation $\mathfrak{p}_1 + \mathfrak{p}_2 +$, $\mathfrak{p}3 = 1$, we get the unique solution $\mathfrak{p} = (0.2926 \quad 0.4634 \quad 0.2440)$ which is the stationary distribution and indeed the limiting distribution. It is the same as the one obtained in Example A3.7.

A3.2.5 Reversible Markov Chains

Often, it is of interest to construct Markov chains with a given limiting distribution $\mathfrak{p}$ and eventually to sample the process states. This is indeed the essence of Markov chain Monte Carlo (MCMC) sampling methods. In this context, we need to highlight the usefulness of reversible Markov chains (Kelly 1979, Aldous and Fill 2002, Strook 2005). A Markov chain with transition matrix $\mathcal{T}$ is called reversible if there exists a probability distribution $\mathfrak{p}$ such that:

$$\mathfrak{p}_i \mathcal{T}_{ij} = \mathfrak{p}_j \mathcal{T}_{ji}, \forall i, j \tag{A3.30}$$

The last equation is known as the detailed balance equation. If such a $\mathfrak{p}$ exists, it is also the stationary distribution of the chain. This is proved below.

Proof: For fixed $i \in \{0, m\}$ where m is the cardinality†† of the state space S, summation on both sides of Equation (A3.30) over $j \in \{0, m\}$ gives:

$$\sum_j \mathfrak{p}_i \mathcal{T}_{ij} = \sum_j \mathfrak{p}_j \mathcal{T}_{ji} \tag{A3.31}$$

But the LHS of the last equation gives rise to:

$$\sum_j \mathfrak{p}_i \mathcal{T}_{ij} = \mathfrak{p}_i \sum_j \mathcal{T}_{ij} = \mathfrak{p}_i \tag{A3.32}$$

since the sum of the elements of each row in $\mathcal{T}$ is unity. Thus, from Equation (A3.31) we get:

$$\mathfrak{p}_i = \sum_j \mathfrak{p}_j \mathcal{T}_{ji}, \forall i, j$$

$$\Rightarrow \mathfrak{p} = \mathfrak{p}\mathcal{T} \tag{A3.33}$$

So $\mathfrak{p}$ is the stationary distribution. The proof indicates that a reversible Markov chain is ergodic.

** Linear dependence
A set of vectors $\{v_1, v_2, ..., v_n\}$ is linearly dependent if there exist numbers $a_1, a_2, ..., a_n$ not all equal to zero such that:

$$a_1 v_1 + a_2 v_2 + \cdots + a_n v_n = 0 \tag{i}$$

In case Equation (i) has only the trivial solution $a_1 = a_2 = \cdots = a_n = 0$, then the set $\{v_1, v_2, ..., v_n\}$ is linearly independent.

†† *Cardinality*
Cardinality of a set is a measure of its size, i.e. the number of elements in the set.

As the name indicates, a reversible Markov chain possesses the following property:

$$P\left(X_0 = a_0, X_1 = a_1, \ldots, X_n = a_n\right)$$

$$= P\left(X_0 = a_n, X_1 = a_{n-1}, \ldots, X_n = a_0\right) \tag{A3.34}$$

The property says that, for a reversible Markov chain with $\left(a_0, a_1, \ldots, a_m\right) \in S^{m+1}$ and a limiting distribution $\mathfrak{p}$, the distribution of the process is the same when it is run backward as when it is run forward.

Assuming $0 \le n \le m$, we prove Equation (A3.34) by writing the LHS as:

$$\begin{aligned}
\text{LHS} &= P\left(X_0 = a_0, X_1 = a_1, \ldots, X_n = a_n\right) \\
&= P\left(X_0 = a_0\right) P\left(X_1 = a_1 \mid X_0 = a_0\right) P\left(X_2 = a_2 \mid X_1 = a_1\right) \ldots \\
&\quad P\left(X_n = a_n \mid X_{n-1} = a_{n-1}\right) \\
&= (\mathfrak{p}_0 P_{01}) P_{12} \ldots P_{n-1,n} \\
&= (\mathfrak{p}_1 P_{10}) P_{12} \ldots P_{n-1,n} \\
&= P_{10} (\mathfrak{p}_1 P_{12}) \ldots P_{n-1,n} \\
&= P_{10} (P_{21} \mathfrak{p}_2) \ldots P_{n-1,n}
\end{aligned} \tag{A3.35}$$

Proceeding further finally leads to:

$$\begin{aligned}
\text{LHS} &= P_{10} P_{21} \ldots P_{n,n-1} \mathfrak{p}_n \\
&= \mathfrak{p}_n P_{n,n-1} P_{n-1,n-2} \ldots P_{10} \\
&= P\left(X_0 = a_n\right) P\left(X_1 = a_{n-1} \mid X_0 = a_n\right) P\left(X_2 = a_{n-2} \mid X_1 = a_{n-1}\right) \ldots \\
&\quad P\left(X_n = a_0 \mid X_{n-1} = a_1\right) \\
&= P\left(X_0 = a_n, X_1 = a_{n-1}, \ldots, X_n = a_0\right) \\
&= \text{RHS}
\end{aligned} \tag{A3.36}$$

In the MCMC method, a Markov chain is designed to be ergodic so that the probability distribution over S asymptotically converges to the target distribution $\mathfrak{p}$. In

achieving this goal, the detailed balance equation (A3.30) is utilized. This aspect is further highlighted in the section to follow.

A3.3 MARKOV CHAIN MONTE CARLO (MCMC) SAMPLING TECHNIQUES

MCMC methods constitute a class of Monte Carlo sampling algorithms that use the theory of Markov chains. Metropolis and Metropolis-Hastings algorithms belong to this category and are widely used to sample target *pdf*s as illustrated in the paragraphs that follow. Application of the algorithm is demonstrated in Section 3.4.2, Chapter 3, while presenting the simulated annealing technique of optimization.

These methods use the theory of Markov chains (Section A3.2) to realize samples from a given target density $f_X(x)$ or to evaluate integrals of the type in Equation (A3.8). The basic idea is to construct a reversible Markov chain having a state space S and whose limiting distribution is the target *pdf*. Thus, the Markov chain is designed to be ergodic in that the probability distribution over S converges to $f_X(x)$ regardless of the initial state. Compared to the Metropolis algorithm, the Metropolis-Hastings one is more general in constructing a reversible Markov chain. Suppose that we are given a target probability distribution $f_X(x)$. The Metropolis-Hastings (MH) algorithm is described below to generate a Markov chain on the state space $S = \{x_0, x_1, \ldots\}$ with $f_X(x)$ as its limiting distribution. It is finally needed to prove that the generated Markov chain is reversible.

A3.3.1 Metropolis-Hastings (MH) Algorithm

At any i^{th} step of the algorithm, let the current state of the RV be $X = x_i$. Now, it involves sampling from a proposal (conditional) density $q(x_j \mid x_i)$ denoted by q_{ij} which is easy to simulate. This proposal density $q(.\,|\,.)$ can be arbitrary but has enough scattering to lead to wide exploration of the support of $f(.)$.

We generate $y \sim q(x \mid x_i)$. Then we update X as:

$$X = x_{i+1} = y \text{ with probability } \alpha(x_i, x_j)$$

$$= x_i \text{ with probability } 1 - \alpha(x_i, x_j) \tag{A3.37a}$$

where $j = i + 1$ and $\alpha(x_i, x_j)$ is known as the acceptance probability and is given by:

$$a(x_i, x_j) \min\left\{1, \frac{f_j}{f_i} \frac{q_{ij}}{q_{ji}}\right\} \tag{A3.37b}$$

Here $f_i = f(x_i)$ and $f_j = f(x_j)$. The updating step in Equation (A3.55a) is accomplished by generating a random number u from the RV $U \sim U(0,1)$ and accepting X_j if $u < \alpha(x_i, x_j)$. Otherwise, we reject the new state and set $X_{i+1} = X_i$.

The transition density matrix of the MH algorithm is $\left[\mathcal{T}_{ij}\right]=\left[q_{ij}\alpha\left(x_i,x_j\right)\right]$. It is required to know how the MH algorithm generates a Markov chain whose limiting distribution converges to $\mathbb{f}(x)$. In this regard, it suffices to show that the algorithm satisfies the detailed balance equation (A3.30), i.e., the resulting Markov chain is reversible.

Proof: The detailed balance equation is:

$$\mathbb{f}_i\mathcal{T}_{ij} = \mathbb{f}_j\mathcal{T}_{ji}, \forall i,j \tag{A3.38}$$

For any i, j, we have:

$$\begin{aligned}\text{LHS} = \mathbb{f}_i\mathcal{T}_{ij} &= \mathbb{f}_i q_{ij}\alpha(x_i,x_j)\\ &= \mathbb{f}_i q_{ij}\min\left\{1,\frac{\mathbb{f}_j}{\mathbb{f}_i}\frac{q_{ij}}{q_{ji}}\right\}\\ &= \min\{\mathbb{f}_i q_{ij},\ \mathbb{f}_j q_{ji}\}\end{aligned} \tag{A3.39a}$$

Similarly it is clear that the RHS of Equation (A3.38) may be shown to be:

$$\mathbb{f}_j\mathcal{T}_{ji} = \min\{\mathbb{f}_j q_{ij}, \mathbb{f}_i q_{ij}\} \tag{A3.39b}$$

The last two equations indicate that the MH algorithm satisfies the detailed balance Equation (A3.30) and hence the Markov chain generated is reversible and the limiting *pdf* of the chain is the target density $\mathbb{f}(x)$.

Example A3.9. Consider simulation of a bimodal *pdf* (Andrieu *et al.* 2003):

$$\mathbb{f}_X(x) = 0.3\,exp(-0.2x^2) + 0.7\,exp(-0.2(x-10)^2) \tag{A3.40}$$

Solution. We use the proposal density $q\left(x|x_i\right)=\frac{1}{10\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{x-x_i}{10}\right)^2\right)$ which is a normal distribution $\mathcal{N}\left(x_i,10\right)$. Note that the selected proposal density is symmetric about x_i, the current state. At the start of computations, x_0 is taken as 1.0. At the i^{th} step, we sample y from $q(x \mid x_i)$ and get x_{i+1} as per the acceptance probability from Equation (A3.37a). The histogram of the samples obtained at the end of 5000 steps is shown in Figure A3.12. The result well approximates the target *pdf* in Equation (A3.40).

One characteristic feature of the MH algorithm is that the sequence x_i may not change for many computational steps because of rejections of the new samples y with probability $1-\alpha\left(x_i,x_j\right)$. However, the consequent retentions of the current states

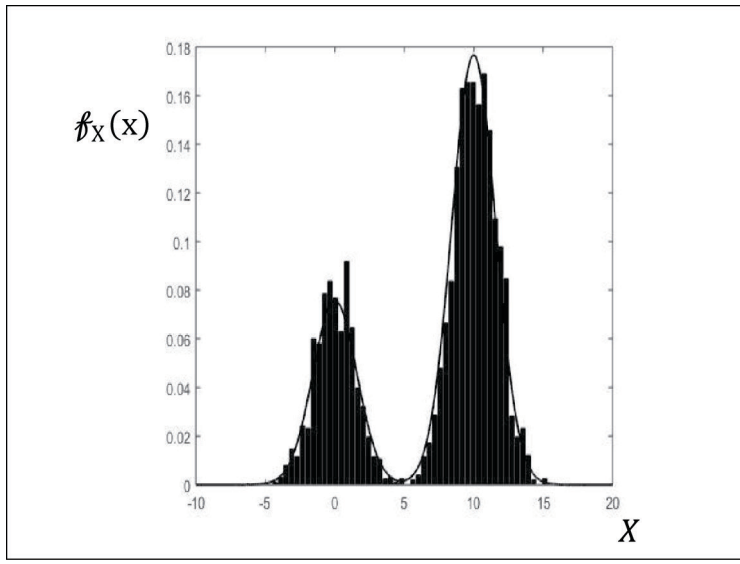


FIGURE A3.12 Sampling of a bimodal *pdf* in Example A3.9, histogram along with the target *pdf* in red (using symmetric proposal *pdf*).

(unlike the rejection method) in the chain indeed renders the sampling of the target *pdf* more efficient. Further, in Example A3.9, the proposal density $q\left(x|x_i\right)$ is symmetric, i.e. $q\left(x_j|x_i\right) = q\left(x_i|x_j\right) \Rightarrow q_{ji} = q_{ij}$ and therefore, the acceptance probability is decided only by $\frac{f_j}{f_i}$ and is independent of q. In fact, the algorithm using a symmetric proposal density is known as the Metropolis algorithm. Choices of symmetric q include normal distributions or uniform distributions centred at the current state x_i. The simulated annealing (SA) method of optimization described in Section 3.4.2, Chapter 3, uses the Metropolis-Hastings algorithm with an unsymmetric proposal *pdf*. The following example illustrates the use of the unsymmetric q.

Example A3.10. We simulate the bimodal *pdf* in the last example by using a uniform distribution $\mathrm{U}(-10,20)$ as a proposal *pdf*. It is asymmetrically used at each i^{th} step with the new state y sampled without centring it at the current state x_i. The target *pdf* $f_X(x)$ is predominantly supported in the chosen interval $[-10,20]$.

Solution. The sampled *pdf* is shown in Figure A3.13 in the form of a histogram. The results obtained by the MH algorithm with 10,000 samples show better convergence than with 5000 samples. The target pdf is also plotted in the figure.

One important application of the MH algorithm is in sampling a distribution when it is known only up to a constant. Such a distribution is usually called an unnormalized *pdf*. We may express an unnormalized *pdf* by $cf(x)$ where c is an unknown constant and $f(x)$ the true normalized *pdf*. Note that the MH algorithm depends only on π

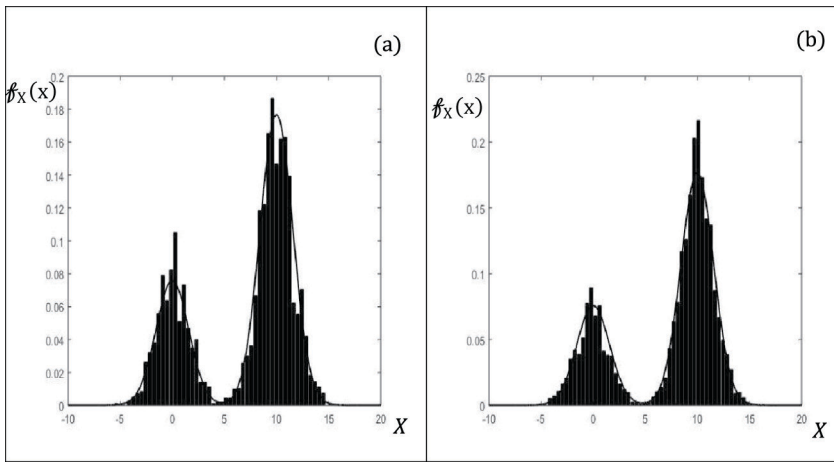


FIGURE A3.13 Sampling of a bimodal *pdf*, histogram along with the target *pdf* in red – using asymmetric proposal *pdf*: (a) histogram drawn with 5000 samples and (b) histogram drawn with 10,000 samples.

through the ratio $\frac{f(x_j)}{f(x_i)} = \frac{\pi(x_j)/c}{\pi(x_i)/c} = \frac{\pi(x_j)}{\pi(x_i)}$ (see Equation A3.37b). In Example 3.9 in Chapter 3, the MH algorithm uses the unnormalized *pdf* corresponding to the Boltzmann distribution without the need to know the normalizing constant c. This aspect is particularly useful in sampling posterior distributions that arise in Bayesian applications (Bernado 1979).

A3.4 ASYMPTOTIC PROPERTY OF MLE $\hat{\theta}$ – FOR LARGE n, $\hat{\theta}$ APPROACHES A NORMAL DISTRIBUTION $\mathcal{N}(\theta, I^{-1/2})$

For large n, $\hat{\theta} \sim \mathcal{N}(\theta, I^{-1/2})$ where I is the Fisher information matrix (FIM). For a proof, we need two basic concepts in probability theory: (i) law of large numbers (LLN) and (ii) central limit theorem (CLT).

(i) LLN is about the asymptotic nature of probabilities of events, empirically estimated from observations/experiments. It states that, for a sequence of independent random variables $X_j, j = 1,2,\ldots,n$ each with a finite mean m, and for any $\varepsilon > 0$:

$$P\left(\left|\frac{X_1 + X_2 + \ldots + X_n}{n} - m\right| \geq \varepsilon\right) \to 0 \text{ as } n \to \infty \tag{A3.41}$$

It implies that for a large number n of trials of a random experiment, the empirical (also called sample or statistical) mean $\frac{X_1 + X_2 + \ldots + X_n}{n}$ often stays close to the value m. Also, see Section A3.1.

ii) CLT states that given n independent random variables $X_j, j = 1,2,\ldots,n$ with a uniform mean m and variance σ^2 and $Y = \frac{X_1 + X_2 + \ldots + X_n}{n}$ another random variable, one has:

$$\frac{Y - m}{\sigma/\sqrt{n}} \sim \mathcal{N}(0,1) \text{ as } n \to \infty \tag{A3.42}$$

where $\mathcal{N}(0,1)$ stands for a standard normal random variable with mean equal to zero and standard deviation of unity.

See Papoulis (1991) and Roy and Rao (2017) for more details.

A3.4.1 Proof for the Asymptotic Property of MLE

With $l'(\boldsymbol{\theta};\boldsymbol{Z}) = \left(\frac{\partial l(\boldsymbol{\theta};\boldsymbol{Z})}{\partial \boldsymbol{\theta}}\right)^T$, $\boldsymbol{I}'^{(\boldsymbol{\theta};\boldsymbol{Z})} = \left(\frac{\partial l(\boldsymbol{\theta};\boldsymbol{Z})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial l(\boldsymbol{\theta};\boldsymbol{Z})}{\partial \boldsymbol{\theta}}\right)^T$ one has by Taylor expansion about the MLE $\hat{\boldsymbol{\theta}}$:

$$0 = l'\left(\hat{\boldsymbol{\theta}};\boldsymbol{Z}\right) = l'(\boldsymbol{\theta};\boldsymbol{Z}) + l''(\boldsymbol{\theta};\boldsymbol{Z})\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right)^T + \text{remainder term} \tag{A3.43}$$

Here $l''(\boldsymbol{\theta};\boldsymbol{Z})$ is the Hessian matrix $\boldsymbol{H}(\boldsymbol{\theta}) \in \mathbb{R}^{m \times m}$ of the log-likelihood function.

(a) From Equation (3.14b) in Chapter 3, we have $l'(\boldsymbol{\theta};\boldsymbol{Z}) = \sum_{i=1}^{n} \frac{\partial}{\partial \boldsymbol{\theta}}\left(\log f_{Z_i}\left(\boldsymbol{z_i};\boldsymbol{\theta}\right)\right)$ and from Equation (3.17):

$$E_{\boldsymbol{Z}}\left[l'(\boldsymbol{\theta};\boldsymbol{Z})\right] = 0 \tag{A3.44}$$

The covariance matrix of $l'(\boldsymbol{\theta};\boldsymbol{Z})$ may be written as:

$$E_{\boldsymbol{Z}}\left[l'(\boldsymbol{\theta};\boldsymbol{Z})l'(\boldsymbol{\theta};\boldsymbol{Z})^T\right] = -E_{\boldsymbol{Z}}\left[\boldsymbol{H}(\boldsymbol{\theta})\right] \text{(from Equation 3.21)} \tag{A3.45}$$

By CLT, as $n \to \infty$, the score function approaches a normal distribution, i.e.:

$$l'\left(\hat{\boldsymbol{\theta}}; \boldsymbol{Z}\right) \to \mathcal{N}\left(0, \boldsymbol{I}\left(\boldsymbol{\theta}\right)^{1/2}\right) \text{(see definition of FIM,, } \boldsymbol{I}\left(\boldsymbol{\theta}\right) \text{ in Equation 3.16)} \tag{A3.46}$$

which is a multi-normal *pdf*.‡‡

(b) Now, by LLN:

$$E_Z\left[l''\left(\boldsymbol{\theta}; \boldsymbol{Z}\right)\right] = E_Z\left[\boldsymbol{H}\left(\boldsymbol{\theta}\right)\right]_{n\to\infty} \tag{A3.47}$$

Thus, knowing the asymptotic properties of $l'\left(\boldsymbol{\theta}; \boldsymbol{Z}\right)$ and $l''\left(\boldsymbol{\theta}; \boldsymbol{Z}\right)$ from Equations (A3.46) and (A3.47), respectively, and ignoring the remainder term in Equation (A3.43), we have:

$$\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right) \sim \boldsymbol{I}^{-1}\left(\boldsymbol{\theta}\right)\mathcal{N}\left(0, \boldsymbol{I}\left(\boldsymbol{\theta}\right)^{1/2}\right) = \mathcal{N}\left(0, \boldsymbol{I}\left(\boldsymbol{\theta}\right)^{-1/2}\right) \text{(from Equation 3.22)}$$

$$\Rightarrow \hat{\boldsymbol{\theta}} \sim \mathcal{N}\left(\boldsymbol{\theta}, \boldsymbol{I}\left(\boldsymbol{\theta}\right)^{-1/2}\right) \tag{A3.48}$$

A3.5 CONFIDENCE INTERVALS

The confidence interval is an interval estimate which provides a range of values that may likely contain the parameter of interest with a specified probability. A 95% confidence interval means that if the random data are sampled (observed) on numerous occasions and interval estimates are made on each occasion, the resulting intervals would contain the true parameters in 95% of the cases. For instance, let $X \sim \mathcal{N}\left(m_X, \sigma_X\right)$. Then $P\left(-1.96\sigma_X \le X - m_X \le 1.96\sigma_X\right) = \Phi_{\mathbb{Z}}\left(\mathbb{Z} = 1.96\right) - \Phi_{\mathbb{Z}}\left(\mathbb{Z} = -1.96\right) = 0.95$, where $\mathbb{Z} = \dfrac{X - m_X}{\sigma_X} \sim \mathcal{N}\left(0,1\right)$, the standard normal RV and $\Phi_{\mathbb{Z}}(.)$ the corresponding

---

‡‡ *Multi-normal pdf*

Let $\boldsymbol{X} = \left(X_1, X_2, \ldots . X_n\right)^T$ be a joint normal RV with mean vector $\boldsymbol{m} = \left(m_1, m_2, \ldots, m_n\right)^T$, and covariance matrix $\boldsymbol{C} = \left[C_{jk}\right]_{n\times n}$ where $m_j = E\left[X_j\right]$ and elements of the covariance matrix $C_{jk} = E\left[\left(X_j - m_j\right)\left(X_k - m_k\right)\right]$, $1 \le j,k \le n$. Then $\boldsymbol{X}$ follows the n-dimensional multi-normal *pdf*:

$$f_X\left(x\right) = \frac{1}{\left(2\pi\right)^{n/2}\sqrt{\Delta}} \exp\left\{-\left(\frac{1}{2}\boldsymbol{X}\boldsymbol{C}^{-1}\boldsymbol{X}^T - \boldsymbol{m}\boldsymbol{X}\right)\right\} \tag{i}$$

where Δ stands for the determinant of the covariance matrix $\boldsymbol{C}$, assumed to be positive definite.

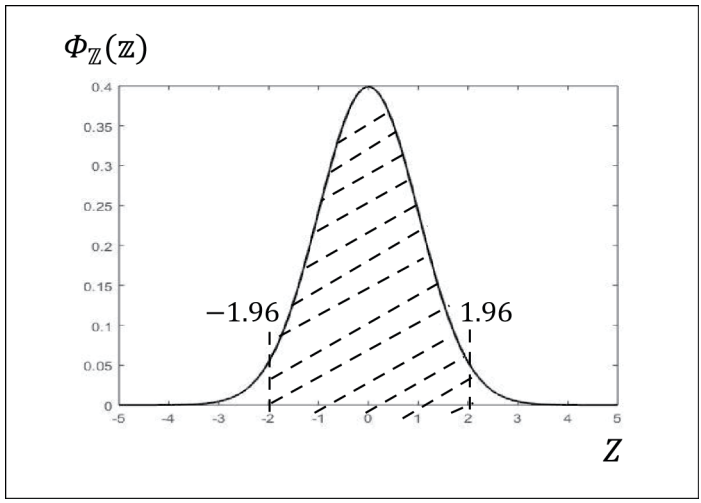


FIGURE A3.14 Standard normal *pdf* $\Phi_{\mathbb{Z}}(\mathbb{Z})$ probability $P(-1.96 \le \mathbb{Z} \le 1.96) = 0.95$ = area of the hatched portion under the *pdf* curve.

CDF. See Figure A3.14 showing the standard normal *pdf* $\Phi_{\mathbb{Z}}(\mathbb{Z})$ along with the interval in which X lies with probability 0.95.

A3.6 GRAM-SCHMIDT ORTHOGONALIZATION PROCEDURE (MEIROVITCH 1980)

Gram-Schmidt orthogonalization is used to construct new vector(s) which is (are) orthogonal ($\perp$) to given vector(s). It is a deflation procedure by which a unit vector, say, $\boldsymbol{A}_1$, is deflated of the given vector $\boldsymbol{d}_1$ so that the new vector $\bar{\boldsymbol{A}}_1$ is orthogonal to $\boldsymbol{d}_1$, i.e., $\bar{\boldsymbol{A}}_1^T \boldsymbol{d}_1 = 0$ (Figure A3.15a). In Figure A3.15b, the vector $\boldsymbol{A}_1$ is deflated of two vectors, $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$.

A3.7 RESONANCES IN A DYNAMICAL SYSTEM

It is known that resonances occur in a system when any of its natural frequencies are close to the frequency of a harmonic input. A resonant state is characterized by large amplitudes of vibration. It is explained here with the example of a spring-mass-damper oscillator model (Figure A3.16), which is usually known to as a single degree of freedom (SDOF) system. The degree of freedom refers to the displacement of the mass point in the system which is free to vibrate under the external input. The SDOF system is governed by the following DE in dynamic equilibrium:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = p(t) \tag{A3.49}$$

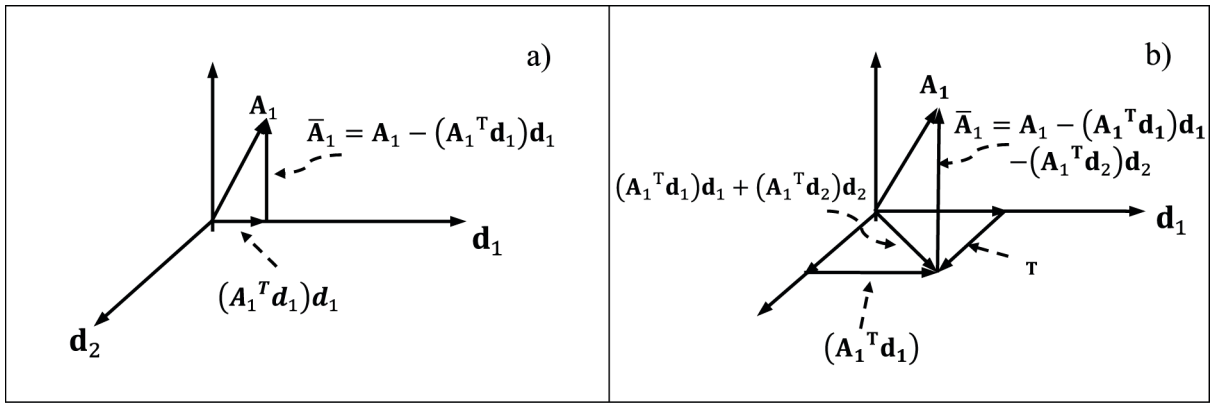


FIGURE A3.15 Gram-Schmidt orthogonalization procedure: (a) $\bar{A}_1 \perp \boldsymbol{d}_1$ and (b) $\bar{A}_1 \perp (\boldsymbol{d}_1 \cap \boldsymbol{d}_2)$.

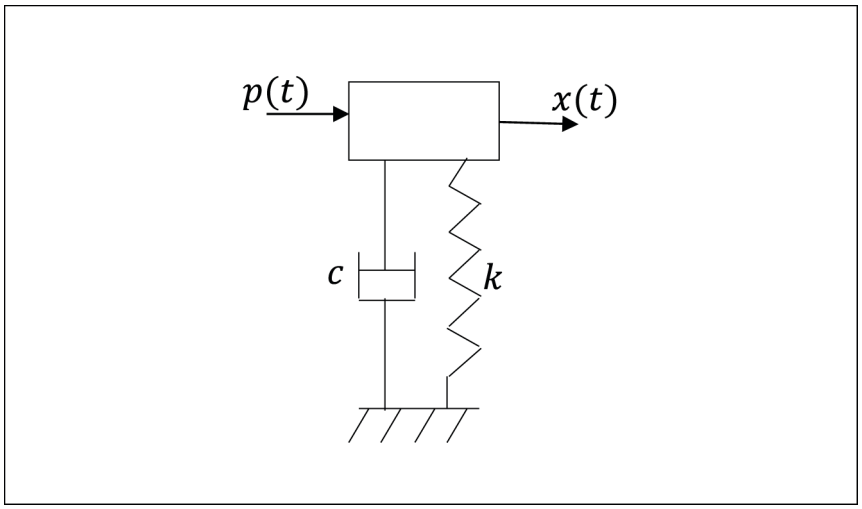


FIGURE A3.16 Single degree of freedom (SDOF) oscillator.

m, c and k are the mass, damping and stiffness parameters, respectively. This is a linear time invariant (LTI) system where the coefficients of the DE above are invariant with time. $x(t)$ is the displacement at the mass point. $p(t)$ is the external input. Under any general input $p(t)$, the solution to Equation (A3.49) is given by (Clough and Penzien 1982):

$$x(t) = e^{-\xi\omega_n t}\left(A\sin\omega_d t + B\ \cos\omega_d t\right) + \int_{-\infty}^{\infty} h(\tau)\, p(t-\tau)\, d\tau \qquad \text{(A3.50)}$$

A and B are the integration constants to be derived from the initial conditions $x(t) = x_0$ and $\dot{x}(t) = \dot{x}_0$. $\omega_n = \sqrt{k/m}$ is the natural frequency of the oscillator. $\xi = c/2\sqrt{km}$ is the damping ratio. $h(t)$ is the impulse response of the system. While the first term on the RHS in the last equation is the transient part of the solution that tends to zero as time $t \to \infty$, the second term is the familiar convolution integral. The impulse response $h(t)$ may be obtained as:

$$h(t) = \frac{1}{m\omega_d}\exp(-\xi\omega t)\sin\omega_d t \qquad \text{(A3.51)}$$

$\omega_d = \omega\sqrt{1-\xi^2}$ is the damped natural frequency. Under a harmonic input $p(t) = A\sin\lambda t$, the steady-state part of the solution may be obtained as:

$$x_s(t) = \frac{A}{k}\frac{\sin(\lambda t - \phi)}{\left\{(1-r^2)^2 + (2\xi r)^2\right\}^{1/2}},\ \phi = \tan^{-1}\left(\frac{2\xi r}{1-r^2}\right) \qquad \text{(A3.52a,b)}$$

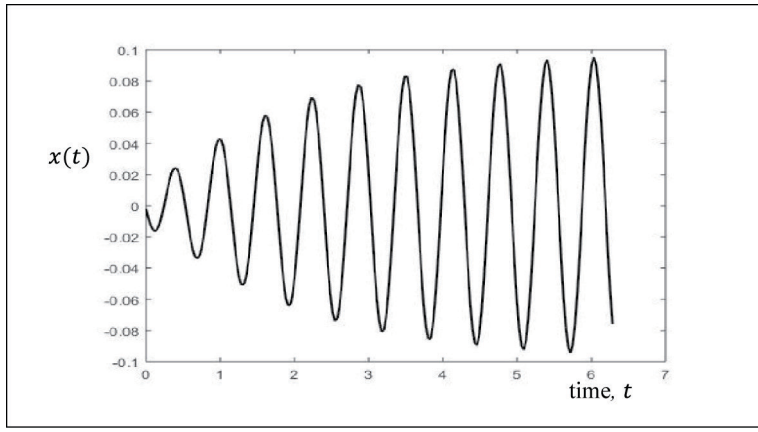


FIGURE A3.17 Unbounded solution to the SDOF oscillator at resonance: $r = \dfrac{\lambda}{\omega_n} = 0.99 \approx 1$.

$r = \dfrac{\lambda}{\omega_n}$ is known as the frequency ratio. Figure A3.17 shows the solution in Equation (A3.50) when $r \approx 1$ with zero initial conditions. The solution of the oscillator may build up unbounded (with or without damping) with time unless it is controlled by a suitable mechanism. This state of solution is called resonance.

A3.8 NATURAL FREQUENCIES AND FREQUENCY RESPONSE

Frequency response is the steady-state response of a linear time-invariant (LTI) system to a harmonic input of the form $p(t) = A\sin\lambda t$. For an SDOF oscillator, Equation (A3.50) gives total solution consisting of both the transient $x_t(t)$ and steady-state $x_s(t)$ parts of the solution. The convolution integral $\int_{-\infty}^{\infty} h(\tau)\,p(t-\tau)\,d\tau$ in this equation yields $x_s(t)$. With the transient part eventually going to zero for large time, $x(t) = x_s(t) = h(t) * p(t)$ by familiar notation. By Fourier inverse transform, $p(t-\tau)$ can be expressed as:

$$p(t-\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} P(j\omega)\,e^{j\omega(t-\tau)}\,d\omega \tag{A3.53}$$

Substituting in the convolution integral, we have:

$$x(t) = x_s(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} h(\tau)\int_{-\infty}^{\infty} P(j\omega)\,e^{j\omega(t-\tau)}\,d\omega\,d\tau$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} h(\tau)\,e^{-j\omega\tau}\,d\tau\right) P(j\omega)\,e^{j\omega t}\,d\omega \tag{A3.54}$$

The inner integral in the last equation is the Fourier transform $H(j\omega)$ of the impulse response function $h(t)$ and $x(t)$ may be expressed as:

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(j\omega)\ P(j\omega)e^{j\omega t}d\omega \tag{A3.55}$$

If we denote $H(j\omega)P(j\omega)$ by $X(j\lambda)$, the above integral shows that $X(j\lambda)$ is the FT of the response $x(t)$ in the frequency domain. Thus, $X(j\omega) = H(j\omega)P(j\omega)$. The FT $H(j\omega)$ of $h(t)$ is generally called the complex frequency function and is given by:

$$H(j\omega) = \frac{1}{(k - m\omega^2) + j2\xi c} \tag{A3.56}$$

$H(j\omega)$ is a function of the system parameters. For a harmonic input $p(t) = A\sin\lambda t$' $P(j\omega) = \frac{A}{2}e^{j(\omega-\lambda)} + \frac{A}{2}e^{j(\omega+\lambda)}$. Therefore, the amplitude and phase of $X(j\omega)$ are given by:

$$|X(j\omega)| = \frac{A}{2}|H(-j\lambda)| + \frac{A}{2}|H(j\lambda)| = A|H(j\lambda)| = \frac{A}{k}\frac{1}{\left\{(1-r^2)^2 + (2\xi r)^2\right\}^{1/2}}$$

$$\arg(X(j\omega)) = \arg(H(j\lambda) = \tan^{-1}\left(\frac{2\xi r}{1-r^2}\right) \tag{A3.57a,b}$$

where $r = \frac{\omega}{\omega_n}$ is the ratio of excitation frequency to the oscillator natural frequency. Figure A3.18 shows the plot of $|X(j\omega)|$ as a function of r. At $r \approx 1$, i.e. when $\omega \approx \omega_n$, the frequency response amplitude plot assumes the maximum peak amplitude which is the resonance point for the oscillator. One may also experimentally obtain the natural frequency $\boldsymbol{\omega}_n$ by exciting the system with a harmonic input, varying its frequency λ and identifying the resonance point.

A practical system such as the shaft in Figure 3.17 in Chapter 3 is continuous in the distribution of its mass and stiffness thus having, in general, infinity of *dof*s. This is unlike the oscillator shown in Figure A3.11 which is a discrete one with only one *dof* and thus having only one natural frequency. The shaft in Figure 3.17a in Chapter 3 is semi-discretized by the FEM (with respect to the spatial coordinates) and the number of *dof*s is reduced from infinity to 64 (number of dof/node × number of nodes in the FE model – Figures 3.17b–c). Analogous to Equation (A3.56), the frequency response for the shaft can be obtained from the FT of Equation (3.43b) in Chapter 3:

$$\boldsymbol{X}(j\omega) = \left[(\boldsymbol{K} - \omega^2\boldsymbol{M}) + j\omega\boldsymbol{C}\right]^{-1}\boldsymbol{\mathcal{P}}(j\omega) \tag{A3.58}$$

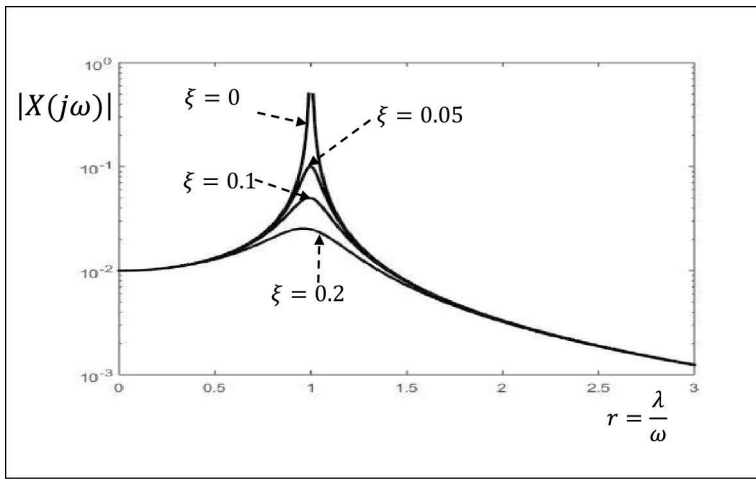


FIGURE A3.18 Frequency response of an SDOF oscillator for different damping ratios $\xi = 0, 0.05, 0.1$ and 0.2, resonance at $r = 1$.

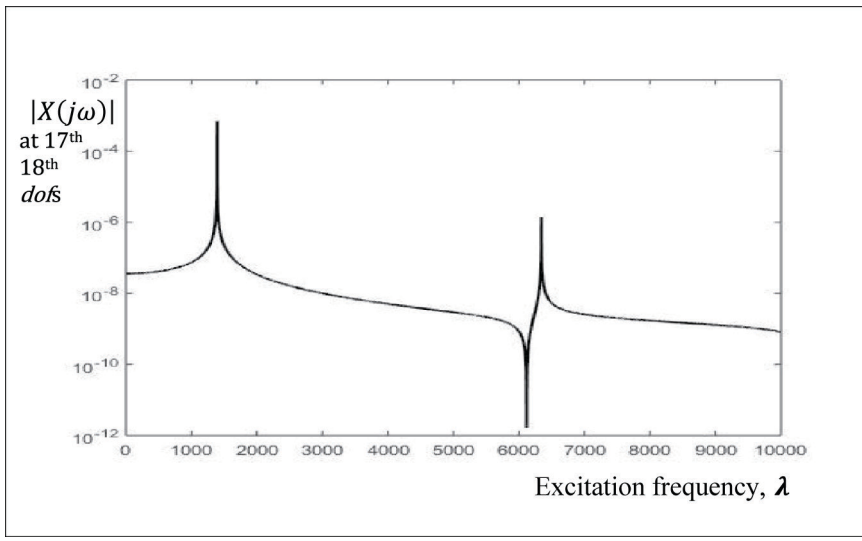


FIGURE A3.19 Frequency response of the spring supported circular shaft in Figure 3.17 in Chapter 3; the figure shows response $|\boldsymbol{X}(\boldsymbol{j\omega})|$ at only two *dof*s, 17 and 18, corresponding to the Y- and Z-directions of the left support point (node 5) of the shaft (both responses are identical).

$\boldsymbol{X}(j\omega), \boldsymbol{\mathcal{P}}(j\omega) \in \mathbb{R}^n$. The frequency response is computed from the last equation at one of the support points on the shaft for the two *dof*s in *Y*- and *Z*-directions and shown in Figure A3.19. The first few natural frequencies of the shaft are identifiable from the peaks from the figure.

REFERENCES

Aldous, D. and J. Fill. 2002. *Reversible Markov Chains and Random Walks on Graphs*. Unfinished monograph.

Andrieu, C, N. De Freitas, N. A. Doucet and M. I. Jordan. 2003. An introduction to MCMC for machine learning. *Machine Learning* 50: 5–43.

Bernado, J. M. 1979. Reference Posterior Distributions for Bayesian Inference. *Journal of the Royal Statistical Society. Series B (Methodological)* 41(2): 113–147.

Clough, R. W. and J. Penzien. 1982. *Dynamics of Structures*. McGraw-Hill.

Dimov. L. T. 2008. *Monte Carlo Methods for Applied Scientists*. London: World Scientific.

Kelly, F. P. 1979. *Reversibility and Stochastic Networks*. John Wiley & Sons Ltd., Chichester. Wiley Series in Probability and Mathematical Statistics.

Meirovitch, L. 1980. *Computational Methods in Structural Dynamics*. Sijthoff and Noordhoff International Publishers. Netherlands.

Norris, J. R. 2012. *Markov Chains*. Cambridge University Press.

Papoulis, A. 1991. *Probability, Random Variables and Stochastic Processes*. 3rd edn. New York: McGraw-Hill, Inc.

Roy, D. and G. V. Rao. 2017. *Stochastic Dynamics, Filtering and Optimization*. Cambridge University Press.

Rubinstein, B. Y. 1981. *Simulation and the Monte Carlo Method*. New York: John Wiley & Sons.

Rudin, W. 1976. *Principles of Mathematical Analysis*. 3rd edn. New York: McGraw-Hill.

Strook, D. W. 2005. *An Introduction to Markov Processes*. Berlin, Heidelberg: Springer Verlag.

von Neumann, J. 1951. Various techniques used in connection with random digits. Monte Carlo methods. *National Bureau of Standards* 12: 36–38.

Appendix 4

A4.1 CHRISTOFFEL SYMBOLS Γ^k_{ij} IN TERMS OF THE SPHERICAL COORDINATES

The Christoffel symbols Γ^k_{ij} are given by:

$$\Gamma^k_{ij} = \frac{1}{2} g^{kl}\left(\frac{\partial g_{jl}}{\partial u^i} + \frac{\partial g_{il}}{\partial u^j} - \frac{\partial g_{ij}}{\partial u^l}\right) \tag{A4.1}$$

where $u^1 = \phi$ and $u^2 = \theta$. With the matrix $g = \begin{bmatrix} 1 & 0 \\ 0 & \sin^2\phi \end{bmatrix}$ and $\begin{bmatrix} 1 & 0 \\ 0 & \operatorname{cosec}^2\phi \end{bmatrix}$, we get:

$$\Gamma^1_{11} = \frac{1}{2}\left\{g^{11}\left(\frac{\partial g_{11}}{\partial \phi} + \frac{\partial g_{11}}{\partial \phi} - \frac{\partial g_{11}}{\partial \phi}\right) + g^{12}\left(\frac{\partial g_{12}}{\partial \phi} + \frac{\partial g_{12}}{\partial \phi} - \frac{\partial g_{11}}{\partial \theta}\right)\right\}$$

$$= 0$$

$$\Gamma^1_{12} = \frac{1}{2}\left\{g^{11}\left(\frac{\partial g_{21}}{\partial \phi} + \frac{\partial g_{11}}{\partial \theta} - \frac{\partial g_{12}}{\partial \phi}\right) + g^{12}\left(\frac{\partial g_{22}}{\partial \phi} + \frac{\partial g_{12}}{\partial \theta} - \frac{\partial g_{12}}{\partial \theta}\right)\right\}$$

$$= 0 = \Gamma^1_{21}$$

$$\Gamma^1_{22} = \frac{1}{2}\left\{g^{11}\left(\frac{\partial g_{21}}{\partial \theta} + \frac{\partial g_{21}}{\partial \theta} - \frac{\partial g_{22}}{\partial \phi}\right) + g^{12}\left(\frac{\partial g_{22}}{\partial \theta} + \frac{\partial g_{22}}{\partial \theta} - \frac{\partial g_{22}}{\partial \theta}\right)\right\}$$

$$= \frac{1}{2}(-2\sin\phi\cos\phi) = -\sin\phi\cos\phi$$

$$\Gamma^2_{11} = \frac{1}{2}\left\{g^{21}\left(\frac{\partial g_{11}}{\partial \phi} + \frac{\partial g_{11}}{\partial \phi} - \frac{\partial g_{11}}{\partial \phi}\right) + g^{22}\left(\frac{\partial g_{12}}{\partial \phi} + \frac{\partial g_{12}}{\partial \phi} - \frac{\partial g_{11}}{\partial \theta}\right)\right\}$$

$$= 0$$

$$\Gamma^{2}_{12} = \frac{1}{2}\left\{\mathfrak{g}^{21}\left(\frac{\partial \mathfrak{g}_{21}}{\partial \phi} + \frac{\partial \mathfrak{g}_{11}}{\partial \theta} - \frac{\partial \mathfrak{g}_{12}}{\partial \phi}\right) + \mathfrak{g}^{22}\left(\frac{\partial \mathfrak{g}_{22}}{\partial \phi} + \frac{\partial \mathfrak{g}_{12}}{\partial \theta} - \frac{\partial \mathfrak{g}_{12}}{\partial \theta}\right)\right\}$$

$$= \text{cosec}\,\phi \cos\phi = \Gamma^{2}_{21}$$

$$\Gamma^{2}_{22} = \frac{1}{2}\left\{\mathfrak{g}^{21}\left(\frac{\partial \mathfrak{g}_{21}}{\partial \theta} + \frac{\partial \mathfrak{g}_{21}}{\partial \theta} - \frac{\partial \mathfrak{g}_{22}}{\partial \phi}\right) + \mathfrak{g}^{22}\left(\frac{\partial \mathfrak{g}_{22}}{\partial \theta} + \frac{\partial \mathfrak{g}_{22}}{\partial \theta} - \frac{\partial \mathfrak{g}_{22}}{\partial \theta}\right)\right\} \tag{A4.2}$$

$$= 0$$

A4.2 MATRIX *g* CORRESPONDING TO RIEMANNIAN METRIC (IN EXAMPLE 4.6) IN TERMS OF LOCAL COORDINATES

The objective function is the Rosenbrock function:

$$f\left(x_1, x_2\right) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2 \tag{A4.3}$$

The matrix elements in gare given by:

$$g_{11} = 1 + \left(\frac{\partial f}{\partial x_1}\right)^2 = 1 - \left\{400 x_1\left(x_2 - x_1^2\right) + 2\left(1 - x_1\right)\right\}^2$$

$$g_{12} = \left(\frac{\partial f}{\partial x_1}\right)\left(\frac{\partial f}{\partial x_2}\right) = -\left\{400 x_1\left(x_2 - x_1^2\right) + 2\left(1 - x_1\right)\right\}\left\{200\left(x_2 - x_1^2\right)\right\} = g_{21}$$

and

$$g_{22} = 1 + \left(\frac{\partial f}{\partial x_2}\right)^2 = 1 + \left\{200\left(x_2 - x_1^2\right)\right\}^2 \tag{A4.4a–c}$$

A4.3 STOCHASTIC PROCESSES, STOCHASTIC CALCULUS AND SOLUTION OF SDES

A few classical evolutionary optimization methods based on stochastic search are described in Chapter 3 (Section 3.4). In the description of these methods and their applications, we have mostly utilized some basic concepts of probability theory and random variables and their simulations. Though based on stochastic search, many of these evolutionary optimization methods are hardly founded on a rigorous probabilistic basis. For optimization methods which are ‘intrinsically stochastic’ (in that they

draw upon the tools of stochastic calculus), we need to familiarize the readers with the theory of stochastic processes and a few associated topics of relevance. With this in view, a quick overview of stochastic differential equations (SDEs) and their solution methods is provided in this section. Note that SDEs are the stochastic analogues of ODEs.

Starting with a formal definition of a stochastic process, we proceed to highlight the properties of Brownian motion – a special class of stochastic processes – also known as the Wiener process (Wiener 1923). Brownian motion has numerous practical applications, especially in modelling noise effects in dynamical systems. It is extensively used in the Euclidean setting to model a variety of phenomena – stock price variations in finance market (Karatzas and Shreve 1991), non-equilibrium statistical mechanics (Kubo 1986) and population dynamics (Wu and Hu 2008). One finds an obvious extension of these models to manifolds for possible use in associated optimization problems. For a comprehensive reading on stochastic processes and related subjects, one may refer to Rogers and Williams (2000) and Roy and Rao (2017).

A4.3.1 Stochastic Processes – A Brief Overview

We find uncertainties invariably in all physical phenomena that are known to behave randomly in time and/or space. The typical wind velocity wave forms in Figure 4.3 in Chapter 4 (assumed to be a collection at a location for a short period on different days under 'identical' conditions) show intrinsic random fluctuations in their histories, extreme values and possibly in the frequency content. One may find similar randomness/stochasticity in many other system models – for instance, communication signals (Gray and Davisson 2004, Jhonson 2013), stock price variations in financial markets (Karatzas and Shreve 1991, Vecer 2011) and mechanics (Vanmarcke 1983, Nigam and Narayanan 1994). Such randomly varying functions in time are often described as stochastic processes. Given a probability space $(\Omega, \mathcal{F}, P)$, we may define a stochastic process X as a parametrized family of random variables $X = \{X_t(\omega) : t \geq 0, t \in T, \omega \in \Omega\}$. The indexing parameter t generally refers to time (or a time-like variable), discrete or continuous and taking values in $\mathbb{R}^+$. For a fixed $\omega' \in \Omega$, $X(\omega', t)$ is often called a path of the stochastic process. For a fixed $t' \in T$, $X(\omega, t')$ is a random variable in $(\Omega, \mathcal{F}, P)$. A stochastic process is path-wise continuous at any $t_k \in [0, T]$, if, for almost all $\omega \in \Omega$, $t \to t_k$ implies $X(\omega, t) \to X(\omega, t_k)$. Thus, a process is continuous if, for almost all $\omega \in \Omega$, $X(\omega, .)$ is a continuous function. We may treat a stochastic process either as an ensemble of (possibly infinitely many) trajectories evolving in time or as a collection of random variables sampled at (possibly infinitely many) time instants. A brief exposition on random variables (both scalar and vector), their distributions/density functions and properties such as independence is provided in Appendix 1.

A stochastic process is represented by the set of its finite dimensional distributions (fdds), i.e. by seeking probabilities of the form $P(X(t_1) \in \mathcal{F}_1, X(t_2) \in \mathcal{F}_2, \ldots, X(t_k)$

$\in \mathcal{F}_k\big)$ for any partition of $[0,T]$ with $0 \le t_1 \le t_2 \le \ldots\ldots t_k \le T$. $\mathcal{F}_i, i = 1,2..,k$ are Borel sets[1] in $\mathbb{R}^n$. A stochastic process is typically characterized by all its finite dimensional joint distributions of the form:

$$F_X\left(x_1, x_2, \ldots..x_m; t_1, t_2, \ldots., t_m\right) = P\left(X\left(t_1\right) \le x_1, X\left(t_2\right) \le x_2, \ldots, X\left(t_m\right) \le x_m\right) \tag{A4.5}$$

If the distribution function is differentiable, one gets the joint pdf $f_X\left(x_1, x_2, \ldots..x_m; t_1, t_2, \ldots., t_m\right) = \dfrac{\partial^m F_X\left(x_1, x_2, \ldots..x_m; t_1, t_2, \ldots., t_m\right)}{\partial x_1 \partial x_2, \ldots..\partial x_m}$. The finite dimensional joint distributions are associated with RVs which in turn correspond to the snapshots of the stochastic process at different time instants. In the rest of the appendix, $X(t)$ or X_t is used to denote a stochastic process for the sake of brevity. In practice, we will often have a countable sequence of random variables $\left\{X_{t_j}, j = 0,1,\ldots,n,\ldots\right\}$, $n \in \mathbb{N}$, representing a stochastic process.

A4.3.2 Brownian Motion/Wiener Process

Brownian motion is a continuous time stochastic process and it is often used to describe irregular and animated motion of particles suspended in fluid, first observed by Robert Brown (1827). It was mathematically constructed by Wiener (1923) and this is the reason that Brownian motion is also known as the Wiener process. A Brownian motion may be thought of as the limiting form of a random walk model.

Proof: A random walk is a discrete stochastic process and is represented by the index set t being finite or countable, e.g. $t \in \mathbb{N} = \{1,2...\}$. Thus, a random walk is a finite/countable collection of random variables $\left\{X_{t_i}(\omega) := X_i(\omega), i \in \mathbb{N}\right\}$. Now, refer to the random motion of a particle in a fluid and let $s_j = \pm\sigma$ be the distance travelled, say on $\mathbb{R}$, by the particle due to a bombardment by the fluid particles at each discrete time instant t_j. s_j is a Bernoulli RV like the tossing of a coin with only two possible outcomes and probability measure $P\left(s_j = +\sigma\right) = 1/2$ and $P\left(s_j = -\sigma\right) = 1/2$. The RVs s_j, $j = 1,2,\ldots$ are independent with $E\left[s_j\right] = 0$ and $E\left[s_j^2\right] = \sigma^2$. Thus, if k is the number of movements equal to $+\sigma$ out of n bombardments per unit time and Δt, the time between two successive bombardments, a random walk is described by the total distance travelled by the particle as

$$X_n := X\left(t_n = n\Delta t\right) = \sum_{j=1}^{n\Delta t} s_j = \left(2k - n\right)\sigma. \tag{A4.6}$$

The associated binomial measure[2] given by:

$$P\left(B_n = m\sigma\right) = \binom{n}{k}\frac{1}{2^n} \text{ with } m = \left(2k - n\right) \tag{A4.7}$$

This leads to $E\left[B_n\right] = 0$ and $E\left[B_n{}^2\right] = n\sigma^2 = \frac{t}{\Delta t}\sigma^2$. In the limit as $n \to \infty$(*or equivalently as* $\Delta t \to 0$), the probability measure approaches the Gaussian (by DeMoivre–Laplace limit theorem – see Papoulis 1991) and the probability distribution takes the form of the standard normal distribution:

$$P\left(B_n \le m\sigma\right) \to F_Z\left(z = m/\sqrt{n}\right) \tag{A4.8}$$

where $Z \sim \mathcal{N}\left(0,1\right)$. In the limit as $\Delta t \to 0$, we assume that $\sigma \approx o\left(\sqrt{\Delta t}\right) \to 0$ so that $E\left[B_t{}^2\right]$ remains finite and $\to \alpha t$ where $\alpha = \frac{\sigma^2}{\Delta t}$. Now, with $\ell = m\sigma$, we have $\frac{m}{\sqrt{n}} = \frac{\ell}{\sqrt{\alpha t}}$ and the one-dimensional probability density of the limiting Brownian motion (Wiener process) $B_t\left(\omega\right)$ is given by:

$$f_B\left(\ell\right) = \frac{1}{\sqrt{2\pi\alpha t}}\exp\left(-\ell^2/2\alpha t\right), \text{i.e.} B \sim \mathcal{N}\left(0, \alpha t\right) \tag{A4.9}$$

■

If B_t denotes (a scalar valued) Brownian motion $B\left(t\right)$ with $B\left(0\right) = 0$, the random variables $\left\{B_{t_i} := B_i\right\}, i \in \mathbb{N}$, $(0 = t_0 < t_1 < t_2 \ldots)$ are zero mean normal. The incremental random variables $B_1 - B_0$, $B_2 - B_1$, etc. are also zero mean normals and, in addition, are independent. It is thus a Gaussian stochastic process with independent increments. A stochastic process $B_t\left(\omega\right)$, $t \ge 0$ adapted to the filtration[3] $\mathcal{F}_t$ is called an $\mathcal{F}_t$-measurable standard Brownian motion if:

$$1.\ B_0 = 0 \text{a.s.}$$

$$2.\ \text{The random variable } B_t - B_s \sim \mathcal{N}\left(0, \sqrt{t-s}\right) \text{ for all } s \le t \tag{A4.10}$$

Thus, a (standard) Brownian motion evolves with zero mean at any time instant and with a variance equal to $t - s$ that increases with time. Since a Brownian

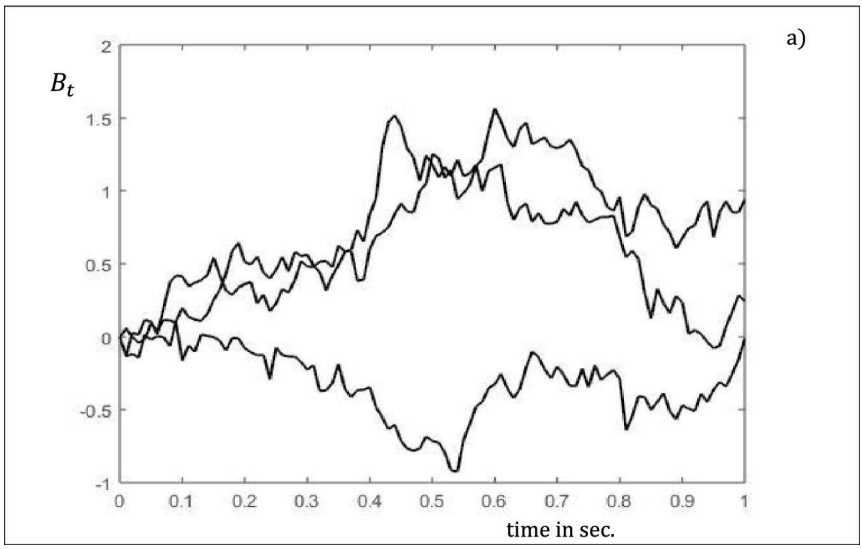


FIGURE A4.1a Brownian motion B_t; a few typical trajectories/paths.

motion is normal, the finite dimensional distribution of the random variables $\left\{B\left(t_j\right) := B_j\right\}, j = 1,2,\ldots,n$, is jointly normal. Note that, given a jointly normal RV $\boldsymbol{X}$ with the pdf:

$$\not{f}_{\boldsymbol{X}}(\boldsymbol{x};t) = (2\pi)^{-n/2}\exp\{-0.5(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{C}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\} \tag{A4.11}$$

where $\mu = \left(\mu_1,\mu_2,\ldots,\mu_n\right)^T$ is the mean vector with components $\mu_j = E[X_j], j = 1,2,...,n$ and $\boldsymbol{C}$ is the covariance matrix with $C_{lm} = E\left[\left(X_l-\mu_l\right)\left(X_m-\mu_m\right)\right]$ $l,m = 1,2,\ldots,n$. Note that $\boldsymbol{C}$ is positive definite and symmetric:

$$\boldsymbol{C} = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{12} & C_{22} & \cdots & C_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ C_{1n} & C_{2n} & \cdots & C_{nn} \end{bmatrix} \tag{A4.12}$$

The properties in Equation (A4.10) lead to a simple algorithm to generate a standard Brownian motion. That is, with $B_0 = 0$, we have $B_{n\Delta t} = B_{(n-1)\Delta t} + \mathcal{N}\left(0,\sqrt{\Delta t}\right)$.

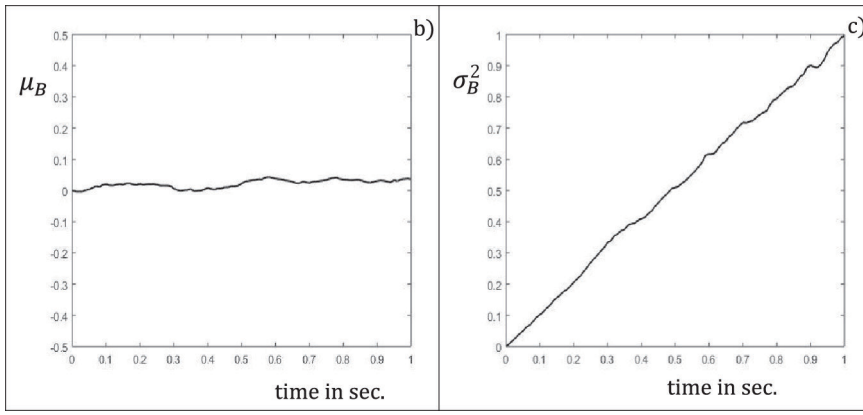


FIGURE A4.1b–c Brownian motion: (b) ensemble mean and (c) ensemble variance over 1000 samples.

Figure A4.1a shows some typical Brownian motion sample paths generated with $\Delta t = 0.01$ and over the interval $t = [0,1]$.

The process mean $\mu_B(t) = E[B_t]$ and variance $\sigma_B^2(t) = E\left[\left(B_t - E[B_t]\right)^2\right]$ are computed over 1000 sample paths and it may be observed from Figures A4.1b–c that the ensemble mean is almost zero and ensemble variance linearly varies (subject to sampling errors) with time.

Note that the Brownian motion is a process with unbounded variation. In this connection, it is useful to define the 'quadratic variation' of a stochastic process X_t, denoted by $[X,X](t)$, as:

$$[X,X](t) := \lim_{\Delta_N \to 0} \sum_{j=0}^{N-1} \left(X_{t_{j+1}} - X_{t_j}\right)^2 \tag{A4.13}$$

where the limit on the RHS envisages a partition Π_N of the interval $[0,t]$ given by $0 = t_0 < t_1 < \ldots < t_N = t$ and $\Delta_N = \max_{1 \le j \le N} \left(t_{j+1} - t_j\right)$, $j = 0,1,\ldots$. Now, if we denote $B_{t_{j+1}} - B_{t_j}$ by ΔB_j and $t_{j+1} - t_j$ by Δt_j, then $\Delta B_j \sim \mathcal{N}\left(0, \sqrt{\Delta t_j}\right)$ and the quadratic variation of Brownian motion is found to be unbounded with time:

$$[B,B](t) = \lim_{\Delta_N \to 0} \sum_{j=0}^{N-1} \left(B_{t_{j+1}} - B_{t_j}\right)^2 = t \tag{A4.14}$$

Proof for $[B,B](t) = \lim_{\Delta_N \to 0} \sum_{j=0}^{N-1} \left(B\left(t_{j+1}\right) - B\left(t_j\right)\right)^2 = t$

Letting $Q_N = \sum_{j=0}^{N-1}\left(B\left(t_{j+1}\right)-B\left(t_j\right)\right)^2$, we have:

$$E\left[Q_N\right] = E\left[\sum_{j=0}^{N-1}\left(B\left(t_{j+1}\right)-B\left(t_j\right)\right)^2\right]$$

$$= \sum_{j=0}^{N-1} E\left[\left(B\left(t_{j+1}\right)-B\left(t_j\right)\right)^2\right]$$

$$= \sum_{j=0}^{N-1}\left(t_{j+1}-t_j\right) = t-0 = t \tag{A4.15}$$

Since the fourth moment of $\mathcal{N}\left(0,\sigma\right)$ is $3\sigma^4$, we have:

$$\text{var}\left(Q_N - t\right) = \text{var}\left(\sum_{j=1}^{N-1}\left(B\left(t_j\right)-B\left(t_{j-1}\right)\right)^2 - t\right) = \sum_{j=1}^{N-1}\text{var}\left(B\left(t_j\right)-B\left(t_{j-1}\right)\right)^2$$

$$= \sum\nolimits_{j=1}^{N-1} 3\left(t_j - t_{j-1}\right)^2 \le 3t\ \max\left(t_j - t_{j-1}\right) = 3t\Delta_N \tag{A4.16}$$

Thus, clearly, $\lim_{\Delta_N\to 0}\text{var}\left(Q_N - t\right) = 0$. This shows that $Q_N - t$ is non-random as $N\to\infty$ and $\left[B,B\right](0,t) = \lim_{N\to\infty} Q_N \to t$ in $L^2\left(P\right)$. Note that even though the sums involved in the definition of Q_N are random, the limit is non-random.

■

Indeed, an interesting property of the squared increment of B_t is its deterministic character which is evident from the proof. More general variants of this identity in terms of deterministic functions multiplying B_t and deterministic functions of Brownian motion are:

$$\lim_{\Delta_N\to 0}\sum_{j=0}^{N-1}\psi_{t_j}\left(\Delta B_j\right)^2 = \int_0^t \psi(s)\,ds$$

$$\text{or, } \lim_{\Delta_N\to 0}\sum_{j=0}^{N-1}\psi\left(B_{t_j}\right)\left(\Delta B_j\right)^2 = \int_0^t \psi\left(B_s\right)ds \tag{A4.17}$$

A4.3.3 Brownian Motion, Though Continuous, Is Not Differentiable Anywhere

Since every increment $\Delta B_s = B_t - B_s \sim \mathcal{N}\left(0, \sqrt{t-s}\right)$, there exists a continuous version of the Brownian motion. But it is not differentiable anywhere. This can be shown to be true by the simple means of testing the convergence of $\lim_{h\to 0} \frac{\Delta B_t(h)}{h} = \lim_{h\to 0} \frac{B_{t+h} - B_t}{h}$ with h being an interval in $\mathbb{R}$. The limit is not continuous and hence the derivative of the Brownian motion does not converge in any sense.

Proof: *Brownian motion is not differentiable anywhere.*

To show this, consider $\frac{\Delta B_t(h)}{h} = \frac{B_{t+h} - B_t}{h}$ with h being an interval in $\mathbb{R}$. $\frac{\Delta B_t(h)}{h}$ is a zero-mean random variable with $E\left[\left(\frac{\Delta B_t(h)}{h}\right)^2\right] = \frac{1}{h}$. If $\lim_{h\to 0} \frac{\Delta B_t(h)}{h}$ converges in some sense to a limit, then the sequence of the characteristic functions $E\left[e^{i\lambda \frac{\Delta B_t(h)}{h}}\right]$ converges to a limit which must be continuous in λ. We know that for a zero-mean normal random variable Z, the characteristic function $\Phi_Z(\lambda) = E\left[e^{i\lambda z}\right] = e^{-\frac{\lambda^2 \sigma_z^2}{2}}$ where σ_z^2 is the variance of Z. Hence, we have:

$$\lim_{h\to 0} E\left[e^{i\lambda \frac{\Delta B_t(h)}{h}}\right] = \lim_{h\to 0} e^{\frac{-\lambda^2}{2h}} = 1, \quad \text{if } \lambda = 0$$

$$= 0, \quad \text{otherwise} \tag{A4.18}$$

We find from the above equation that the characteristic function in the limit is not continuous in λ and hence the derivative of the Weiner process does not converge in any sense.

♦

A4.3.4 White Noise Process

By a non-rigorous approach, one may define a white noise process $W(t)$ as a 'generalized' derivative of a Brownian motion $B(t)$. The word 'generalized' is used since a white noise process $W(t)$ is not a derivative in the usual sense; see Roy and

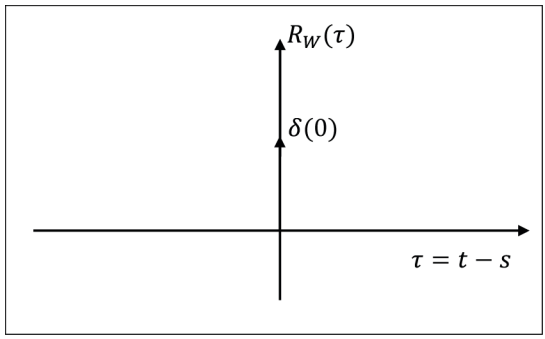


FIGURE A4.2 Autocorrelation function of a white noise process $W(t)$ which is a stationary process.

Rao (2017) for details. This is the Einstein's model (1905) of Brownian motion in his studies on the microscopic motion of particles suspended in fluid and, formally, it is the solution of:

$$\frac{dB}{dt} = W(t) \tag{A4.19}$$

Langevin's alternative model (1908) of Brownian motion is formally the solution of:

$$\frac{d^2B}{dt^2} + c\frac{dB}{dt} = W(t) \tag{A4.20}$$

$c\dfrac{dB}{dt}$ is velocity term denoting random friction forces due to the particles' collisions while $\dfrac{d^2B}{dt^2}$ is the inertia term associated with the particle mass. The white noise process $W_t := W(t)$ is a stochastic process with $E[W_t] = 0$ and $E[W_t W_s] = \delta(t-s)$, where $\delta(.)$ is the Dirac delta function. The Dirac delta property of the autocorrelation function $R_W(t,s) = E[W_t W_s]$ indicates that (i) the RVs W_t and W_s are independent for every $t \neq s$ and (ii) it is a stationary stochastic process. The definition of a stationary process is given in Section A4.3.13. Figure A4.2 shows the autocorrelation function of a white noise process. White noise models find wide applications in applied sciences and engineering and myriad other fields (Cohen 2005, Boyat and Joshi 2015, Azizi and Yazdi 2019).

A4.3.5 Brownian Motion Is a Markov Process

A Brownian motion or Wiener process B_t possesses the Markovian property.[4] A Markov process is characterized by the transition probability function defined as:

$$P(y,t \mid x,s) = P(X(t) \le y \mid X(s) = x) \tag{A4.21}$$

$P(y,t \mid x,s)$ is the conditional probability distribution function of the process at time t given that it is x at time $s < t$. Since the increment $B_t - B_s \sim \mathcal{N}\left(0, \sqrt{t-s}\right)$ for a Brownian motion, the transition probability function is given by:

$$P(y,t \mid x,s) = P(B_t \le y | B_s = x) = \frac{1}{\sqrt{2\pi(t-s)}} \int_{-\infty}^{y} e^{-\frac{(u-x)^2}{2(t-s)}}\, du \tag{A4.22}$$

The corresponding transition probability density function is:

$$f(y,t;x,s) = \frac{1}{\sqrt{2\pi(t-s)}} e^{-\frac{(y-x)^2}{2(t-s)}} \tag{A4.23}$$

A4.3.6 A Wiener Process Is a Martingale[5]

An $\mathcal{F}_t$-adapted Wiener process B_t is a martingale with respect to $\mathcal{F}_t$ and the proof follows.

Proof: We first note that the Brownian motion B_t is integrable with $E\left[B_t\right] = 0 < \infty$. With $0 \le s < t$, we have:

$$E\left[B_t | \mathcal{F}_s\right] = E\left[B_t - B_s + B_s | \mathcal{F}_s\right]$$

$$= E\left[\left(B_t - B_s\right) | \mathcal{F}_s\right] + E\left[B_s | \mathcal{F}_s\right]$$

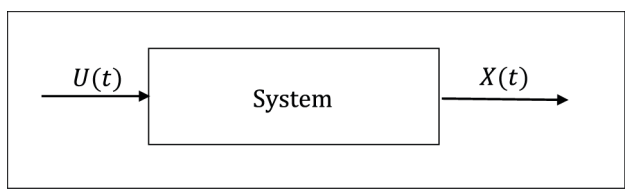


FIGURE A4.3 Dynamical system under external input.

$$= E\left[B_t - B_s\right] + B_s \tag{A4.24}$$

Note that $\mathcal{F}_t = \sigma\left(B_s : s \le t\right)$ and hence $B_t - B_s$ is independent of $\mathcal{F}_s$ leading to $E\left[\left(B_t - B_s\right)|\mathcal{F}_s\right] = E\left[B_t - B_s\right] = 0$. It follows that:

$$E\left[B_t|\mathcal{F}_s\right] = B_s \tag{A4.25}$$

Thus, a Brownian motion is a martingale.

♦

With this brief description of Brownian motion, we proceed to the topic of stochastic differential equations (SDEs), deferring the issue of how these concepts in conjunction with Riemannian geometry are utilized to handle optimization problems for later.

A4.3.7 Ordinary Differential Equations (ODEs) vs. Stochastic Differential Equations (SDEs)

Let us consider an ODE modelling a dynamical system (Figure A4.3) in the state space form:

$$\dot{x} = f\left(x,t\right), x\left(0\right) = x_0 \tag{A4.26}$$

Here $f\left(x,t\right)$ may contain the external input functions $U\left(t\right)$. Now, assuming that the system parameters and the input functions are deterministic, the ODE may be written in the integral form:

$$x\left(t\right) = x_0 + \int_0^t f\left(x,s\right)ds \tag{A4.27}$$

By Picard's iteration, if we start with an initial approximation $\phi_0\left(t\right) = x\left(0\right) := x_0$, we recursively obtain:

$$\phi_{k+1}\left(t\right) = X_0 + \int_0^t f\left(\phi_k\left(s\right),s\right)ds \tag{A4.28}$$

$\lim_{n\to\infty} \phi_n\left(t\right)$ converges to a unique solution $x\left(t\right)$ provided that $f\left(x,t\right)$ is continuous in both arguments X and t and Lipschitz continuous[6] in x.

If either the system parameters or the external inputs are inherently stochastic which is indeed the case in general, the DE governing the system dynamics is known as an SDE. The integral representation in Equation (A4.27) typically contains integrals

where the integrand and integrator may be stochastic processes. Note that for ODEs or SDEs, the system dynamics may be described by integrals of the following kind:

$$I(0,t) = \int_0^t X(s)\,ds$$

$$= \int_0^t X(s)\,dY(s) \qquad \text{(A4.29a,b)}$$

Equation (A4.29a) is the familiar Riemannian integral and (A4.29b) is Riemann-Stieltjes integral. $X(t)$ or/and $Y(t)$ may be stochastic processes in an SDE. In either of the above integral types, the continuity condition of the integrand or the integrator may not be satisfied for an SDE. The reason is that the SDEs may involve, in general, Brownian motion that is not differentiable. Suppose, in general, that an SDE is written in the following differential form:

$$dX(t) = \alpha(t,X)\,dt + \sigma(t,X)\,W(t)\,dt$$

$$= \alpha(t,X)\,dt + \sigma(t,X)\,dB(t) \qquad \text{(A4.30)}$$

That $W(t)$ is a 'generalized' derivative of a Brownian motion $B(t)$ is utilized in writing the second part of the last equation. $\alpha(t,X)$ is known as the drift term and $\sigma(t,X)$ the diffusion term. In integral form, one writes the SDE as:

$$X(t) = X_0 + \int_0^t \alpha\big(s,X(s)\big)\,ds + \int_0^t \sigma\big(s,X(s)\big)\,dB(s) \qquad \text{(A4.31)}$$

The random variable X_0 denotes a measurable initial state of $X(t)$. The first integral on the RHS of the last equation is similar to the one in Equation (A4.29a) and the second one is similar to the integral in Equation (A4.29b). Under these situations, a proper interpretation of the existence of such integrals in Equation (A4.29) must be laid out, in order to make the stochastic analogue of Picard's iteration valid over $[0,T]$ for the SDE in Equation (A4.30). In particular, the task is to properly define the stochastic integral:

$$\mathfrak{T}(X) = \int_0^T X(s)\,dB(s) \qquad \text{(A4.32)}$$

where the upper limit is taken as $T \in \mathbb{R}$. With any partition Π_N of the interval $[a,b]$ given by $a = t_0 < t_1 < \ldots < t_N = b$ and with $\Delta_N = \max_{0 \le j \le N-1}\left(t_{j+1} - t_j\right)$, the integral $\mathfrak{T}(X)$ is approximated in the Riemannian sense (Rudin 1976) by the following limiting sequence:

$$\mathfrak{T} = \lim_{N\to\infty} \sum_{j=0}^{N-1} X(t')\left(B\left(t_{j+1}\right) - B(t_j)\right) \tag{A4.33}$$

In principle, one can create an infinite sequence of such summands corresponding to a choice of $t' \in \left[t_j, t_{j+1}\right] \forall j$ and thus define an approximation to $\mathfrak{T}$ as $\cong \mathfrak{T}' = \sum_{j=0}^{N-1} X_{t'}\left(B_{t_{j+1}} - B_{t_j}\right)$ Thus, the choice of t' matters in defining a stochastic integral (see Section A4.3.14 for a corroboration through examples). Specifically, the integral $\mathfrak{T}$ for $t' = t_j$ is called the Ito integral (Ito 1951). Note that if $t' = \left(t_j + t_{j+1}\right)/2$ (the mid point in $\left[t_j, t_{j+1}\right]$), it leads to another integral representation known as the Stratonovich integral (Stratonovich 1966).

A4.3.8 Existence and Uniqueness of Solution to SDEs

In the case of the ODE (A4.26), a solution exists if $f(X,t)$ is continuous and bounded in a hyper rectangle $R^n : |X - X_0| < r, |t - t_0| < s$. In addition, if $\frac{\partial f}{\partial X}$ is continuous and bounded, the solution is unique. However, the uniqueness is guaranteed if $f(X,t)$ is only Lipschitz continuous (which is a weaker condition than the requirement of $\frac{\partial f}{\partial X}$ being continuous), i.e.:

$$\left|f\left(X_2,t\right) - f\left(X_1,t\right)\right| \le L\left|\left(X_2 - X_1\right)\right| \tag{A4.34}$$

for some positive constant *L*.

Similarly, for the SDE in Equation (A4.30) with the coefficients $\alpha(t,x)$: $[0,\infty]\times\mathbb{R}^m \to \mathbb{R}^m$, $\sigma(t,X) : [0,\infty]\times\mathbb{R}^m \to \mathbb{R}^{m\times n}$ and $E\left[\left|X_0\right|^2\right] < \infty$, a stochastic process $X(t)$ is a unique and time-continuous solution if:

(i) α and σ are measurable functions, uniformly continuous in $t \in [0,T]$ and Lipschitz in *X*, i.e.:

$$\left|\alpha(t,X) - \alpha(t,Y)\right| + \left|\sigma(t,X) - \sigma(t,Y)\right| \le L|X - Y|,\ X,Y \in \mathbb{R}^m \quad \text{and } 0 \le t \le T, \tag{A4.35}$$

for some positive constant L and

(ii) the coefficients satisfy a linear growth condition, i.e.:

$$\left|\alpha(t,X)\right|+\left|\sigma(t,X)\right|\le C\left(1+\left|X\right|\right), X\in\mathbb{R}^m \text{ and } 0\le t\le T \tag{A4.36}$$

for some positive constant C (see Karatzas and Shreve 1991, Klebaner 1998, Øksendal 2003, Roy and Rao 2017 for a comprehensive exposition).

So far, an Ito integral $\mathfrak{T}=\int_0^T X(s)dB(s)$ (similar to the second integral on RHS of Equation A4.31) is defined for a fixed terminal time T. Since $\mathfrak{T}$ is a random variable for any fixed $t\le T$, it immediately follows that $\int_0^t X(s)dB(s)$ as a function of the upper limit t can be considered as a stochastic process – generally known as an Ito process. An Ito process may, in general, be expressed as the integral form of the SDE in Equation (A4.31). The last integral on the RHS of this equation is the Ito integral and the preceding one a non-stochastic integral (despite $\alpha(t,X)$ being possibly stochastic).

From physical considerations, the drift part $\alpha(t,X)$ of the SDE could be a force term derivable from a potential (this is often the case with mechanical oscillators). The diffusion part $\alpha(t,X)$ represents the external 'noise' term that arises from the stochastic environment to which the system is subjected. The noise coefficient σ in the above equation might be system-dependent, i.e. dependent on $X(t)$ also. In this case, the noise is referred to as multiplicative. Otherwise, it is known as additive, i.e., when the diffusion coefficient is just $\sigma(t)$.

A4.3.9 Ito's Formula

Stochastic integration thus defined via Ito integral leads us to a new form of differential calculus applicable to SDEs commonly referred to as 'stochastic calculus'. An insightful treatment of dynamical systems under stochastic loading is possible via a systematic application of stochastic calculus using the Ito integral. The primary tool here is Ito's formula which plays the counterpart of the conventional chain rule in ordinary differential calculus. For example, if $X(t):[0,\infty)\to\mathbb{R}$ is continuous and of bounded variation and $g(X,t)$ a continuously differentiable function of X, then the fundamental theorem of conventional calculus gives:

$$dg\left(X(t),t\right)=\dot{g}\left(X(t),t\right)dt+g'\left(X(t),t\right)dX(t) \tag{A4.37}$$

where $\dot{g}(t)=dg/dt$ and $g'=dg/dX$.

Ito's formula is a stochastic analogue of the above rule when $X(t)$ is a stochastic (Ito) process. Specifically, if $g\left(t,X(t)\right)$ is twice continuously differentiable on $[0,\infty)\times\mathbb{R}$, i.e., $g\in C^2\left([0,\infty)\times\mathbb{R}\right)$, we have Ito's formula:

$$dg\left(t,X_t\right)=\frac{\partial g}{\partial t}\left(t,X_t\right)dt+\frac{\partial g}{\partial x}\left(t,X_t\right)dX_t+\frac{1}{2}\frac{\partial^2 g}{\partial x^2}\left(t,X_t\right)\left(dX_t\right)^2 \tag{A.4.38}$$

Note that, in contrast to Equation (A4.37), Ito's formula contains an additional term containing the second-order derivative of $g(t,X)$. When $X(t)$ is governed by a scalar SDE similar to Equation (A4.30), then we get Ito's formula for $g(t,X_t)$ as:

$$dg(t,X_t)=\left(\frac{\partial g}{\partial t}(t,X_t)+a_t\frac{\partial g}{\partial x}(t,X_t)+\frac{1}{2}\sigma_t^2\frac{\partial^2 g}{\partial x^2}(t,X_t)\right)dt + \sigma_t\frac{\partial g}{\partial x}(t,X_t)dB_t \tag{A4.39}$$

The last result is obtained by substituting for dX_t (from Equation A4.30 into Equation A4.38) and via simplification using the properties of a Brownian motion (Roy and Rao 2017). When $\boldsymbol{X}\in\mathbb{R}^m$, Ito's formula for a scalar function $g(t,\boldsymbol{X}_t)$ is:

$$dg(t,\boldsymbol{X}_t)=\left(\frac{\partial g}{\partial t}(t,\boldsymbol{X}_t)+L_t\left(g(t,\boldsymbol{X}_t)\right)\right)dt+\frac{\partial g}{\partial \boldsymbol{x}}(t,\boldsymbol{X}_t)\boldsymbol{\sigma}_t d\boldsymbol{B}_t \tag{A4.40}$$

where the operator L_t is known as backward Kolmogorov operator defined as:

$$L_t=\sum_{i=1}^{m}a_i(t,\boldsymbol{X}_t)\frac{\partial}{\partial x_i}+\frac{1}{2}\sum_{i,j=1}^{m}\sum_{l=1}^{n}\sigma_{il}(t,\boldsymbol{X}_t)\sigma_{jl}(t,\boldsymbol{X}_t)\frac{\partial^2}{\partial x_i\partial x_j} \tag{A4.41}$$

Note that, in Equation (A4.40), $\frac{\partial g}{\partial x}(t,\boldsymbol{X}_t)=\left\{\frac{\partial g}{\partial x_1},\frac{\partial g}{\partial x_2},\ldots..,\frac{\partial g}{\partial x_m}\right\}$ is a vector, $\boldsymbol{\sigma}$ is an $m\times n$ dimensional diffusion matrix and $\boldsymbol{B}_t$ is an n-dimensional Brownian motion (with independently evolving scalar Brownian components). The integral form of Equation (A4.40) may be written as:

$$g(t,\boldsymbol{X}_t)=g(0,\boldsymbol{X}_0)+\int_0^t\left(\frac{\partial g}{\partial x}(s,\boldsymbol{X}_s)+L_s\left(g(s,\boldsymbol{X}\boldsymbol{s}_t)\right)\right)ds + \int_0^t\frac{\partial g}{\partial x}(s,\boldsymbol{X}_s)\boldsymbol{\sigma}_s d\boldsymbol{B}_s \tag{A4.42}$$

where the last term (Ito integral) on the RHS of the above equation is a zero-mean martingale (as $\left|\frac{\partial g}{\partial \boldsymbol{x}}\right|$ is bounded owing to the stipulated continuity of g with respect to the components of $\boldsymbol{X}$ – see Roy and Rao [2017] for details). By taking expectations on both sides of Equation (A4.42), we arrive at what is referred to as Dynkin's formula:

$$E\left[g\left(t,\boldsymbol{X}_t\right)\right]=g\left(0,\boldsymbol{X}_0\right)+E\int_0^t\left(\frac{\partial g\left(s,\boldsymbol{X}_s\right)}{\partial s}+L_s\left(g\left(s,\boldsymbol{X}_s\right)\right)\right)ds \qquad \text{(A4.43)}$$

$\boldsymbol{X}_0$ is assumed to be non-random in arriving at the last equation. Note that given an SDE, one may obtain the underlying system statistics using Ito's and Dynkin's formulae. The statistics are the expectations/moments of a required order obtained in terms of the solution $\boldsymbol{X}\left(t\right)$. Solutions of SDEs characterized in terms of these expectations are often referred to as weak solutions which may only be of interest in most cases. On the other hand, some applications may particularly need computation of strong solutions – path-wise solutions – for example, in the stochastic optimization problems. These problems obviously require methods to solve SDEs numerically. We provide in the next section a brief outline on numerical methods for solving SDEs to obtain strong solutions. Rigorous treatment of the topic may be found in Kloeden and Platen (1992) and Roy and Rao (2017).

A4.3.10 Numerical Solutions to SDEs

Deterministic ODEs may be numerically solved to different orders of accuracy, say by classical Taylor expansion (Simmons and Krantz 2006, Roy and Rao 2012). Similar expansions based on an iterated Ito's formula which is the stochastic analogue of the classical Taylor expansion help to construct numerical integration schemes for SDEs.

A4.3.11 Classical Taylor's Expansion for ODEs

Consider a (scalar) ODE of the form:

$$\frac{dx}{dt}=a\left(t,x\right) \qquad \text{(A4.44)}$$

Assuming that $a\left(t,x\right)$ is sufficiently smooth and varies as an at most linear function of x as $x\to\infty$ so that a unique solution $x\left(t\right)$ exists, we get the familiar Taylor expansion of $x\left(t+h\right)$ in a neighbourhood of t in powers of the increment h and in different derivatives of $a\left(t,x\left(t\right)\right)$:

$$x\left(t+h\right)=x\left(t\right)+ha\left(t,x\left(t\right)\right)+\frac{h^2}{2}La\left(t,x\left(t\right)\right)+\ldots+\frac{h^m}{m!}L^{m-1}a\left(t,x\left(t\right)\right)$$

$$+\text{remainder} \qquad \text{(A4.45)}$$

where L is a differential operator:

$$L=\frac{\partial}{\partial t}+a\left(t,x\right)\frac{\partial}{\partial x} \qquad \text{(A4.46)}$$

The truncated expansion excluding the remainder term corresponds to an explicit method with an accuracy of the m^{th} local order.

A4.3.12 Ito-Taylor's Expansion for SDEs

Now, consider a scalar SDE as in (A4.30) driven by Brownian motion $B(t)$:

$$dX(t) = a(t,X)dt + \sigma(t,X)dB(t), 0 \le t \le T, X(0) = X_0 \tag{A4.47}$$

With $t \le s_1, s_2, s_3 \le s = t + h$, repeated applications of Ito's formula (A4.42) yield the following Ito-Taylor expansion for one-step approximation:

$$X(t+h) = X(t) + \sigma\int_t^{t+h} dB(s_1) + a\int_t^{t+h} ds_1 + L_1\sigma\int_t^{t+h}\left(\int_t^{s_1} dB(s_2)\right)dB(s_1)$$

$$+L_0\sigma\int_t^{t+h}\int_t^{s_1} ds_2 dB(s_1) + L_1 a\int_t^{t+h}\int_t^{s_1} dB(s_2)ds_1$$

$$+L_1^2\sigma\int_t^{t+h}\left(\int_t^{s_1}\left(\int_t^{s_2} dB(s_3)\right)dB(s_2)\right)dB(s_1) + (L_0 a)\int_t^{t+h}\int_t^{s_1} ds_2 ds_1$$

$$+ \text{remainder} \tag{A4.48}$$

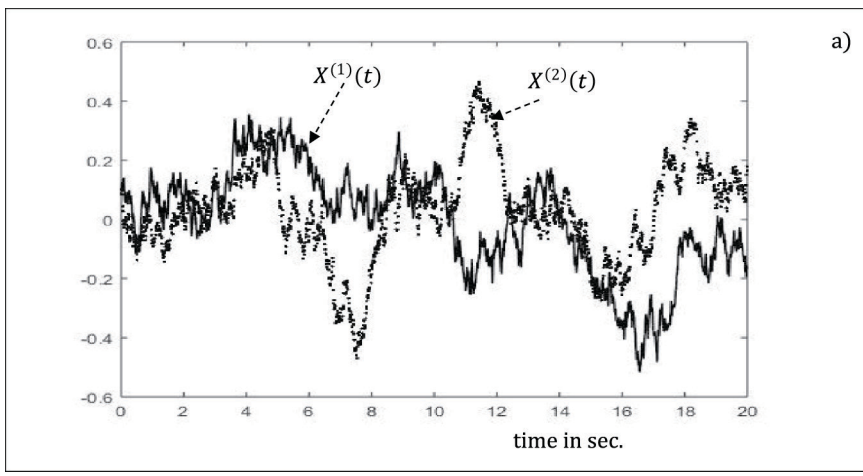


FIGURE A4.4a Numerical solution to the SDE (A4.49) by the EM method; $a = 1.0$ and $\sigma = 0.2$, time step $\Delta t = 0.01$, two solution paths $X^{(1)}(t)$ and $X^{(2)}(t)$ shown by dark lines and dotted lines, respectively.

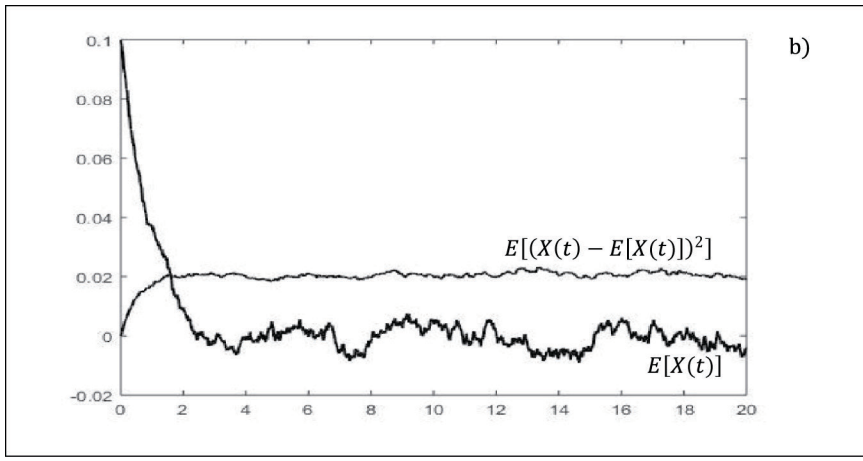


FIGURE A4.4b Ensemble (sample) averages – mean $E\left[X(t)\right]$ and variance $E\left[\left(X(t)-E\left[X(t)\right]\right)^2\right]$ – using 1000 EM-simulated samples from the SDE (A4.49); $a = 1.0$ and $\sigma = 0.2$, time step $\Delta t = 0.01$.

where $L_0 \equiv \frac{\partial}{\partial t} + a\frac{\partial}{\partial x} + \frac{1}{2}\sigma^2\frac{\partial^2}{\partial x^2}$ and $L_1 \equiv \sigma\frac{\partial}{\partial x}$. See Roy and Rao (2017) for a derivation of the last equation and the associated orders of convergence. Depending on the truncation of the one-step approximation above, one gets explicit integration schemes for SDEs with different orders of accuracy. For instance, the expansion in Equation (A4.48), if truncated beyond the first three terms on the RHS gives the explicit Euler-Maruyama (EM) method (1955) with $O\left(h^{\frac{1}{2}}\right)$ global order of convergence (Milstein 1974, Kloeden and Platen 1992). The EM method is the stochastic analogue of the classical Euler method for solving ODEs. One may retain the first five terms in Equation (A4.48) (Milstein 1974) and have an Ito-Taylor scheme of order $O(h)$.

Example A4.1. Let us consider the following one-dimensional SDE and use the EM method to numerically integrate it:

$$dX(t) = -aX(t)dt + \sigma dB(t) \tag{A4.49}$$

Solution. The solution to the SDE (A4.49) is popularly known as the Ornstein–Uhlenbeck process. Two solution trajectories (paths) of $X(t)$ are obtained by the EM method and shown in Figure A4.4a. The parameters a and σ are taken as 1.0 and 0.2, respectively. A step size of $\Delta t = 0.01$ is used. The initial condition X_0 is taken as a deterministic constant equal to 0.1.

Note that it is possible to get weak solutions (moments) for the SDE (A4.49) in closed from. If we apply the expectation operator on both sides of the SDE, we get:

$$E\left[dX(t)\right] = -aE\left[X(t)\right]dt \left(\text{since } E\left[dB(t)\right] = 0\right)$$

$$\Rightarrow dE\left[X(t)\right] = -aE\left[X(t)\right]dt$$

$$\Rightarrow E\left[X(t)\right] = E\left[X(0)\right]e^{-at} \tag{A4.50}$$

As $t \to \infty$, the mean process $E\left[X(t)\right]$ approaches zero. This is also evident from Figure A4.4b where the history is obtained as an ensemble average over 1000 EM-simulated samples. Now, similar to the mean history, the variance history is also obtained as an ensemble average and shown in the same figure. The variance tends to a value 0.02 as $t \to \infty$. In fact, this result may be verified from a closed-form solution. If we take $g(X) = X^2(t)$ and apply Ito's formula (Equation A4.40), one obtains:

$$dg = \left(-2aX^2 + \sigma^2\right)dt + 2\sigma X dB(t)$$

$$\Rightarrow dX^2(t) = \left(-2aX^2 + \sigma^2\right)dt + 2\sigma X dB(t) \tag{A4.51}$$

Applying the expectation operator on both sides of the last equation, one gets:

$$d\left(E\left[X^2(t)\right]\right) = \left(-2aE\left[X^2(t)\right] + \sigma^2\right)dt \left(\text{since } E\left[dB(t)\right] = 0\right) \tag{A4.52}$$

With $E\left[X^2(0)\right] = E\left[X_0^2\right]$, integration once of the ODE (A4.52) gives:

$$E\left[X^2(t)\right] = e^{-2at}\left(E\left[X_0^2\right] + \frac{\sigma^2}{2a}\left(e^{2at} - 1\right)\right)$$

$$= \frac{\sigma^2}{2a} + e^{-2at}\left(E\left[X_0^2\right] - \frac{\sigma^2}{2a}\right) \tag{A4.53}$$

As $t \to \infty$, $E\left[X^2(t)\right] \to \frac{\sigma^2}{2a}$. Since $E\left[X(t)\right] \to 0$ for large t, the variance tends to $\frac{\sigma^2}{2a}$ which is 0.02 for the selected values of a and σ (as also indicated by the variance history in Figure A4.4b obtained by simulation).

An extension of the numerical integration scheme to vector SDEs is straightforward (though the algebra associated with the Ito-Taylor expansion might be cumbersome).

■

Suppose that we have an SDE with $X \in \mathbb{R}^m$ and driven by an n-dimensional Brownian motion:

$$d\boldsymbol{X}(t) = \boldsymbol{a}(t,\boldsymbol{X})dt + \boldsymbol{\sigma}(t,\boldsymbol{X})d\boldsymbol{B}(t), 0 \le t \le T, \boldsymbol{X}(0) = \boldsymbol{X}_0 \tag{A4.54}$$

Here $\boldsymbol{a}(t,\boldsymbol{X}): \mathbb{R}^+ \times \mathbb{R}^m \to \mathbb{R}^m$ and $\boldsymbol{\sigma}(t,\boldsymbol{X}): \mathbb{R}^+ \times \mathbb{R}^m \to \mathbb{R}^{m\times n}$. Following similar steps as in the one -dimensioanl case, one gets the Ito-Taylor expansion for one-step approximation:

$$\begin{aligned}
\boldsymbol{X}(t+h) = \boldsymbol{X}(t) &+ L_0\boldsymbol{X}(t)\int_t^{t+h} ds_1 \\
&+ L_0{}^2\boldsymbol{X}(t)\int_t^{t+h}\int_t^{s_1} ds_2 ds_1 \\
&+ \sum_{k=1}^{n} L_{1k}L_0\boldsymbol{X}(t)\int_t^{t+h}\int_t^{s_1} dB_k(s_2)ds_1 \\
&+ \sum_{k=1}^{n} L_{1k}\boldsymbol{X}(t)\int_t^{t+h} dB_k(s_1) \\
&+ \sum_{k=1}^{n} L_0L_{1k}\boldsymbol{X}(t)\int_t^{t+h}\int_t^{s_1} ds_2 dB_k(s_1) \\
&+ \sum_{k=1}^{n}\sum_{j=1}^{n} L_{1j}L_{1k}\boldsymbol{X}(t)\int_t^{t+h}\int_t^{s_1} dB_j(s_2)dB_k(s_1) \\
&+ \text{remainder}
\end{aligned} \tag{A4.55}$$

Here $L_0{}^2 = L_0 \circ L_0$, i.e. the composition of L_0 with itself. Similar definitions apply to $L_{1k}L_0$, L_0L_{1k} and $L_{1j}L_{1k}$. One may identify different numerical integration schemes of increasing order of accuracy from the expansion.

A4.3.13 Stationary Stochastic Process

Suppose that the random vectors $\left\{X_{t_1}, X_{t_2}, \ldots, X_{t_k}\right\}$ and $\left\{X_{t_{1+\tau}}, X_{t_{2+\tau}}, \ldots, X_{t_{k+\tau}}\right\}$ have the same probability distribution, i.e., the probability density function f is invariant in time-translations, i.e., for any $\tau > 0$, we have:

$$f(x_1, x_2, \ldots, x_k; t_1, t_2, \ldots, t_k) = f(x_1, x_2, \ldots, x_k; t_1 + \tau, t_2 + \tau, \ldots, t_k + \tau) \tag{A4.56}$$

Such a stochastic process is called stationary and the property is referred to as stationarity of order k. The process X is weakly or wide-sense stationary if $k=2$. Weak stationarity implies that $\mu_t = E\left[X_t\right]$ is constant and the covariance $C_{XX}(s,t)$ is a function of only the time difference $t-s$. The covariance function is also known as the autocovariance function and is given by $E\left[\left(X_s-\mu_s\right)\left(X_t-\mu_t\right)\right]$. Note that stationarity of order k implies stationarity of all lower orders.

Example A4.2. We show that $X(t)=A\cos(\lambda t+\theta)$ is a weakly stationary stochastic process where θ is a uniformly distributed RV over $[-\pi,\pi]$ and $A,\lambda\in\mathbb{R}$.

Solution. The first-order moment is:

$$E\left[X(t)\right]=E\left[A\cos(\lambda t+\theta)\right]=A\cos\lambda t E\left[\cos\theta\right]-A\sin\lambda t E\left[\sin\theta\right]$$

$$=A\cos\lambda t\int_{-\pi}^{\pi}\cos\theta f_\theta(\theta)d\theta-A\sin\lambda t\int_{-\pi}^{\pi}\sin\theta f_\theta(\theta)d\theta$$

$$=\frac{A\cos\lambda t}{2\pi}\int_{-\pi}^{\pi}\cos\theta\, d\theta-\frac{A\sin\lambda t}{2\pi}\int_{-\pi}^{\pi}\sin\theta\, d\theta=0 \qquad \text{(A4.57)}$$

Denoting $E\left[X(t)\right]$ by $\mu(t)$, one has:

$$R_{XX}(s,t)=E\left[\left(A\cos(\lambda t+\theta)-\mu(t)\right)\left(A\cos(\lambda s+\theta)-\mu(s)\right)\right]$$

$$=A^2E\left[\cos(\lambda t+\theta)\cos(\lambda s+\theta)\right]$$

(since $\mu(t)=0=\mu(s)$)

$$=A^2\left(\cos\lambda t\cos\lambda s\ E\left[\cos^2\theta\right]+\sin\lambda t\sin\lambda s\ E\left[\sin^2\theta\right]\right)$$

$$-A^2\left(\cos\lambda t\sin\lambda s\ E\left[\cos\theta\sin\theta\right]+\sin\lambda t\cos\lambda s\ E\left[\cos\theta\sin\theta\right]\right)$$

$$=A^2\cos\lambda(t-s) \qquad \text{(A4.58)}$$

The result in the last step is obtained because $E\left[\cos^2\theta\right]=1=E\left[\sin^2\theta\right]$ and $E\left[\cos\theta\sin\theta\right]=0$.

■

A4.3.14 The Choice of t' Matters in Defining a Stochastic Integral $\mathfrak{T}(X) = \int_0^T X(s)dB(s)$

With $\Delta B_j = B_{t_{j+1}} - B_{t_j}$, let us have:

$$\mathfrak{T} = \sum_{j=0}^{N-1} X_{t_j} \Delta B_j$$

$$\overline{\mathfrak{T}} = \sum_{j=0}^{N-1} \overline{X}_{t_{j+1}} \Delta B_j \tag{A4.59a,b}$$

Since X_{t_j} is $\mathcal{F}_{t_j}$-measurable and hence independent of ΔB_j, it follows that:

$$E[\mathfrak{T})] = \sum_{j=0}^{N-1} E\left[X_{t_j} \Delta B_j \right] = 0$$

$$\text{var}(\mathfrak{T}) = E\left[\left(\int_0^T X(s) dB(s) \right)^2 \right] = \sum_{j=0}^{N-1} E\left[X_{t_j}^2 \right] \left(t_{j+1} - t_j \right) \tag{A4.60a,b}$$

However, for $\overline{\mathfrak{T}}$ in Equation (A4.59b), $\overline{X}_{t_{j+1}}$ is not independent of ΔB_j and the moment information for the integral cannot be so easily evaluated. Therefore, the two integrals should be different from each other.

A4.4 TO DRAW SAMPLES OF A GIVEN PROBABILITY DISTRIBUTION: EXAMPLE FOR A SAMPLING PROBLEM

In the example, we solve an over-damped Langevin SDE and generate a Markov chain. The generated sample $x(t)$ is expected to converge to the target *pdf*. converge to a limiting distribution that approximates the target *pdf* $f_X(x)$, i.e. $f_X(x)$. The SDE is of the form:

$$dX_t = -\nabla \log f_X(X)\, dt + \sqrt{2} dB_t \tag{A4.61}$$

In discretized form by EM method:

$$X_{k+1} = X_k - \nabla \log f_X(X_k)\, \Delta t + \sqrt{2}\, \Delta B_k \tag{A4.62}$$

At each time step, Metropolis-Hastings algorithm (Appendix 3) is used to accept or reject the sample X_{k+1}. The scheme is equally applicable for sampling a multivariate *pdf*.

Example A4.3. We sample a bivariate normal $\boldsymbol{X} = \left(X_2, X_2\right)^T$ with mean vector $\boldsymbol{\mu} = \left(2,\ 5\right)^T$ and covariance matrix $\boldsymbol{\Sigma} = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 1 \end{bmatrix}$.

Solution. In this case, the target *pdf* $\boldsymbol{f}_{\boldsymbol{X}}(\boldsymbol{x})$ is:

$$\boldsymbol{f}_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{2\pi|\boldsymbol{\Sigma}|} exp\left(-\frac{1}{2}\ (\boldsymbol{x}-\boldsymbol{\mu})^T\ \boldsymbol{\Sigma}^{-1}\ (\boldsymbol{x}-\boldsymbol{\mu})\ \right) \tag{A4.63}$$

The log likelihood function is:

$$L(\boldsymbol{x}) = \log\left(\boldsymbol{f}_{\boldsymbol{X}}(\boldsymbol{x})\right) = -\log 2\pi - \log|\boldsymbol{\Sigma}| - \frac{1}{2}\ (\boldsymbol{x}-\boldsymbol{\mu})^T\ \boldsymbol{\Sigma}^{-1}\ (\boldsymbol{x}-\boldsymbol{\mu}) \tag{A4.64}$$

and

$$\nabla \log \boldsymbol{f}_{\boldsymbol{X}}(\boldsymbol{x}) = -\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \tag{A4.65}$$

In solving the SDE (A4.61) by the EM method, the discrete map in Equation (A4.62) is similar to maximizing $L\left(x\right)$. During the implementation of the EM method, the first derivative $\nabla \log \boldsymbol{f}_{\boldsymbol{X}}(\boldsymbol{x})$ is obtained by numerical differentiation. In these MCMC methods, it is always preferred to leave out a few initial samples before accepting the rest and this phase goes by the name 'burn-in period'. We simply discard the samples collected during the burn-in and conduct the MH acceptability test at each time step. Figures A4.5–A4.8 show the sampling results for the target *pdf*. The burn-in ratio is 0.2, that is, out of the accepted samples, 20% of the samples are ignored.

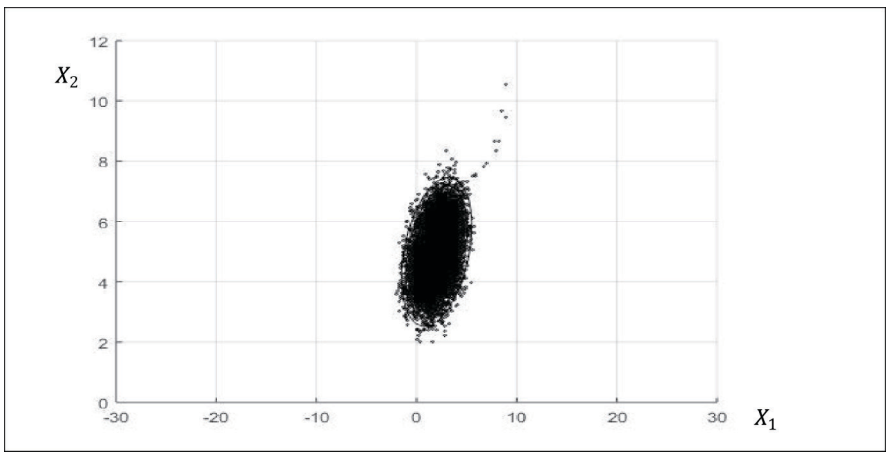


FIGURE A4.5 Sampling of a bivariate Gaussian *pdf*; samples before burn-in; $X_0 = \left(10,10\right)$, $dt = 0.2$, number of samples = 20000, burn-in ratio = 0.2.

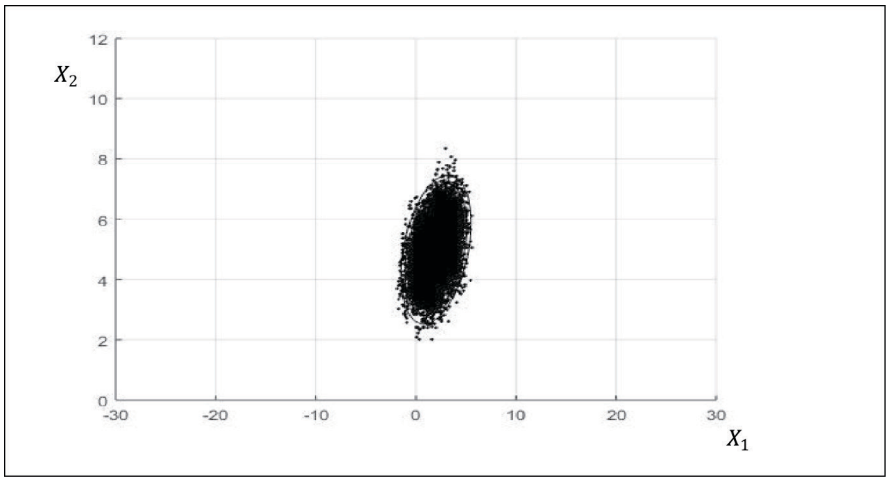


FIGURE A4.6 Sampling of a bivariate Gaussian *pdf*; samples after burn-in; $\boldsymbol{X}_0$ = (10, 10), $dt = 0.2$, number of samples = 20000, burn-in ratio = 0.2.

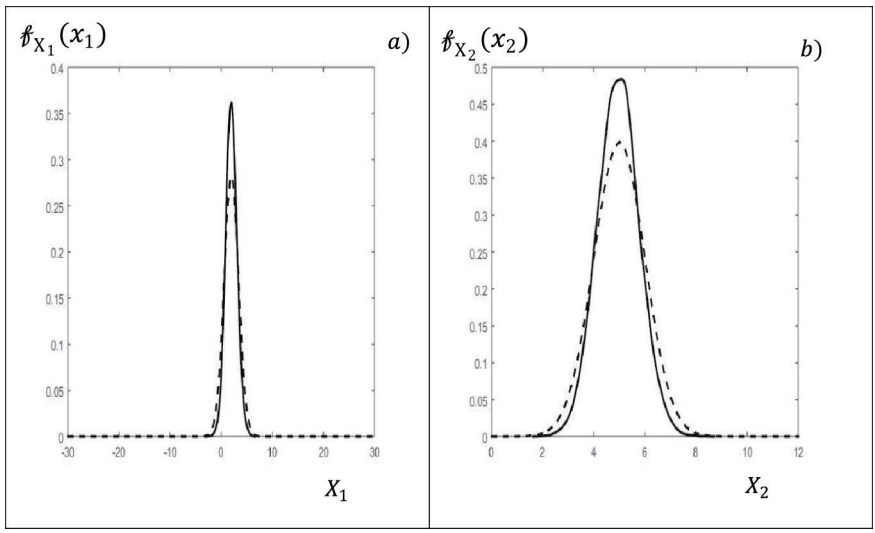


FIGURE A4.7 Sampling of a bivariate Gaussian *pdf*: (a) original and sampled *pdf*s for the first RV and (b) original and sampled *pdf*s for the second RV, $\boldsymbol{X}_0 = (10,10)$, $dt = 0.2$, number of samples = 20,000, burn-in ratio = 0.2, dashed line – original *pdf* and dark line – sampled *pdf*.

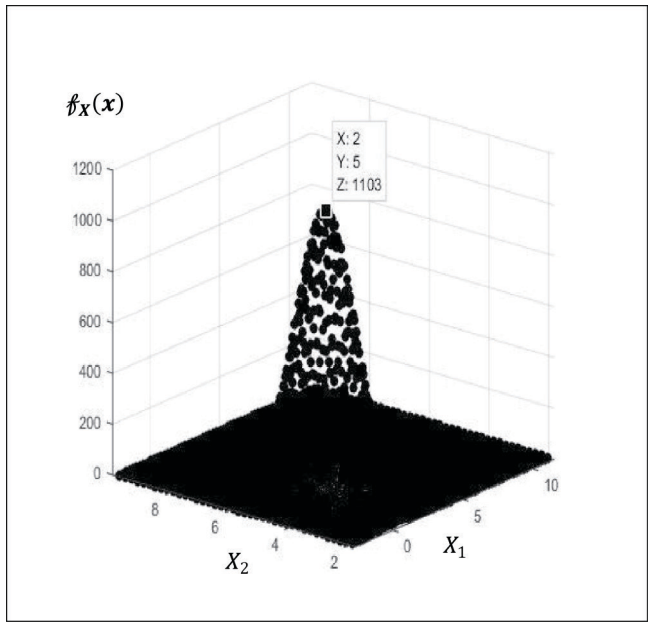


FIGURE A4.8 Sampling of a bivariate Gaussian *pdf*; 3-D plot of the two-dimensional multivariate normal *pdf*, $\boldsymbol{X}_0 = (10,10)$, $dt = 0.2$, number of samples = 20,000, burn-in ratio = 0.2.

■

A4.5 MATRIX $\mathfrak{g}$ AND THE CONNECTION MATRICES FOR THE ACKLEY FUNCTION IN EXAMPLE 4.9

The Ackley function is given in Equation (4.112). Let

$$T_1 = \exp\left(-b\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right), T_2 = \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(cx_i\right)\right) \tag{A4.66}$$

The first-order Euclidean derivative is:

$$\frac{\partial f}{\partial x_i} = -a\frac{\partial T_1}{\partial x_i} - \frac{\partial T_2}{\partial x_i} \tag{A4.67}$$

where

$$dT1 := \frac{\partial T_1}{\partial x_i} = -\frac{bT_1}{\sqrt{n}}\frac{x_i}{\sqrt{\sum_{i=1}^{n}x_i^2}} \text{ and } dT2 := \frac{\partial T_2}{\partial x_i} = -c\frac{T_2}{n}\sin\left(cx_i\right) \tag{A4.68}$$

Matrix $\mathfrak{g}$ *associated with the Riemannian metric g:*

Let $S = \sum_{i=1}^{n} x_i^2$ and *ddT*1 and *ddT*2 stand for the Hessian matrices of T_1 and T_2. Then:

$$\left[ddT1\right]_{j,k} = \frac{bT_1}{\sqrt{n}} x_j x_k \left(\sum_{i=1}^{n} x_i^2\right)^{-1.5} - \frac{bT_1}{\sqrt{n}} \frac{1}{\sqrt{\sum_{i=1}^{n} x_i^2}} \delta_{jk} - \frac{b}{\sqrt{n}} \frac{x_i}{\sqrt{\sum_{i=1}^{n} x_i^2}} \left(dT1\right)_k \text{ and}$$

$$\left[ddT2\right]_{j,k} = -c^2 \frac{T_2}{n} \cos\left(cx_j\right) \delta_{jk} - c \frac{\left(dT2\right)_k}{n} \sin\left(cx_j\right) \qquad \text{(A4.69a,b)}$$

$$\mathfrak{g} = -0.5^* a \left[ddT1\right] - 0.5^* \left[ddT2\right] \qquad \text{(A.4.70)}$$

Matrices corresponding to the Cristoffel symbols:
Let *dddT*1 and *dddT*2 stand for the third derivatives of T_1 and T_2. Then:

$$dddT1\left[j,k,m\right) = \frac{bT_1}{\sqrt{n}} x_k \left(\sum_{i=1}^{n} x_i^2\right)^{-1.5} \delta_{jm} + \frac{bT_1}{\sqrt{n}} x_j \left(\sum_{i=1}^{n} x_i^2\right)^{-1.5} \delta_{km}$$

$$- \frac{3bT_1}{\sqrt{n}} x_j x_k x_m \left(\sum_{i=1}^{n} x_i^2\right)^{-2.5} + \frac{b}{\sqrt{n}} x_j x_k \left(\sum_{i=1}^{n} x_i^2\right)^{-1.5} \left(dT1\right)_m$$

$$+ \frac{bT_1}{\sqrt{n}} x_m \left(\sum_{i=1}^{n} x_i^2\right)^{-1.5} \delta_{jk} - \frac{b}{\sqrt{n}} \left(\sum_{i=1}^{n} x_i^2\right)^{-0.5} \delta_{jk} \left(dT1\right)_m$$

$$+ \frac{b}{\sqrt{n}} x_j x_m \left(\sum_{i=1}^{n} x_i^2\right)^{-1.5} \left(dT1\right)_k - \frac{b}{\sqrt{n}} \left(\sum_{i=1}^{n} x_i^2\right)^{-1.5} \delta_{jm} \left(dT1\right)_k$$

$$- \frac{b}{\sqrt{n}} x_j \left(\sum_{i=1}^{n} x_i^2\right)^{-0.5} \left(ddT1\right)_{km} \qquad \text{(A4.71a)}$$

and

$$dddT2\left[j,k,m\right) = c^3 \frac{T_2}{n} \sin\left(cx_j\right) \delta_{km} - \frac{c}{n} \sin\left(cx_j\right) \left[ddT2\right]_{km}$$

$$- \frac{c^2}{n} \cos\left(cx_j\right) \delta_{jk} \left(dT2\right)_m - \frac{c^2}{n} \cos\left(cx_j\right) \delta_{jm} \left(dT2\right)_k \qquad \text{(A4.71b)}$$

The derivatives of the matrix $\mathfrak{g}$ are given by:

$$\frac{\partial \mathfrak{g}}{\partial x_j}, j = 1,2,\ldots,m = -0.5a\left[dddT1\right] - 0.5*\left[dddT2\right] \tag{A4.72}$$

The Christoffel symbols are given by Equation (A4.1):

$$\Gamma_{ij}^{k} = \frac{1}{2}\mathfrak{g}^{kl}\left(\frac{\partial \mathfrak{g}_{jl}}{\partial u^i} + \frac{\partial \mathfrak{g}_{il}}{\partial u^j} - \frac{\partial \mathfrak{g}_{ij}}{\partial u^l}\right) \tag{A4.73}$$

NOTES

1 Borel sets are the sets that can be constructed from open or closed sets through the operations of countable union and intersection. For example, if $\Omega = (0,1]$, then any open or closed interval in Ω is a Borel set. Note that the collection of all Borel sets on Ω constitutes a Borel σ-algebra. See p. 449.

2 Binomial measure
A binomial measure is the probability measure P corresponding to binomial distribution. Binomial distribution is a discrete probability distribution $B(n,p)$ characterized by two parameters n and p. n represents the number of independent experiments and each experiment is called a binomial trial having only two outcomes, say success or failure, with respective probabilities p and $1-p$. If X denotes the binomial random variable, $X \sim B(n,p)$ with $n \in \mathbb{N}$ and $p = [0,1]$. The probability of getting exactly k successes out of n experiments (or trials) is:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{i}$$

where $\binom{n}{k}$ is the binomial coefficient and is equal to $\frac{n!}{k!(n-k)!}$. See p. 450.

3 Filtration
If, for every $i \in I$, where $I \subset \mathbb{N}$ is an index set starting from 1, $\mathcal{F}_{t_i \le t}$ is a sub-σ-algebra of $\mathcal{F}$ defining a probability space, then filtration is the set $\left\{\mathcal{F}_{t_i}\right\}_{i \in I}$. Intuitively, it is an ordered collections of subsets that are used to model the information available on a random process at a given time. See p. 450.

4 Markovian property
A stochastic process $X(t)$ is Markov if the conditional distribution of $X(t+\tau)$ given $\mathcal{F}_t$ is the same as the conditional distribution of $X(t+\tau)$ given X_t for any time increment $\tau > 0$. In other words,

$$P(X(t+\tau) \le x \mid \mathcal{F}_t) = P(X(t+\tau) \le x \mid X_t) \tag{i}$$

Thus, for a Markov process, a future state is dependent only on the present and not on the past. Markovian property is crucial in characterizing solutions to SDEs. See p. 456.

5 A martingale $\mathcal{M}(t)$ is a special type of $\mathcal{F}_t$ -adapted L^1 (*i.e.*$E\left[\mathcal{M}(t)\right] < \infty$) stochastic process satisfying the condition:

$E\left[\mathcal{M}(t)|\mathcal{F}_s\right] = \mathcal{M}(s)$ a.s. (almost surely) for every $s \le t$ (i)

If $E\left[\mathcal{M}(t)|\mathcal{F}_s\right] \ge \mathcal{M}(s)$ a.s. for every $s \le t$, $\mathcal{M}(t)$ is a sub-martingale.

If $E\left[\mathcal{M}(t)|\mathcal{F}_s\right] \le \mathcal{M}(s)$ a.s. for every $s \le t$, $\mathcal{M}(t)$ is a super-martingale.

Brownian motion is an example for a martingale. A stopped stochastic process is also an example; it is defined as:

$$X_t^\tau = X_t, t < \tau < \infty$$

$$= X_t, t \ge \tau \quad \text{(ii)}$$

Here τ is a random variable with sample space $\Omega = \{0,1,\ldots,\infty\}$. See p. 456.

6 Lipschitz continuity

A function $f(x,t)$ is said to be Lipschitz continuous in the variable x on a set $S \in \mathbb{R}^2$, if there exists a constant $L > 0$ such that:

$$\left|f(x_1,t) - f(x_2,t)\right| \le L\left|x_1 - x_2\right| \quad \text{(i)}$$

whenever both points (x_1,t) and (x_2,t) are in *S*. The constant L is called the Lipschitz constant for *f*. A sufficient condition for Lipschitz condition is the differentiability of $f(x,t)$ in the variable x on the set S. In other words, if:

$$\left|\frac{\partial f(x,t)}{\partial x}\right| \le L, \forall x,t \in S \quad \text{(ii)}$$

then, $f(x,t)$ is Lipschitz continuous. This is verifiable by means of the mean value theorem. See p. 457.

REFERENCES

Azizi, A. and Yazdi, P. G. 2019. *White Noise: Applications and Mathematical Modeling*. Springer Nature Singapore.

Boyat, A. K. and Joshi, B. K. 2015. A review paper: noise models in digital image processing. *Signal & Image Processing: An International Journal* (SIPIJ), 6(2): 63–75.

Brown, R. 1827. *A Brief Account of Microscopical Observations, etc*., London (not published).

Cohen, L. 2005. The history of noise. *IEEE Signal Processing Magazine*, 22(6): 20–45.

Einstein, A. 1905. On the movement of small particles suspended in stationary liquids required by the molecular-kinetic theory of heat. *Annals of Physics*, 17: 549–560. Appearing in *The Collected Papers of Albert Einstein*. English translation by Anna Beck. Princeton, NJ: Princeton University Press, 1989.

Gray, R. M. and Davisson, L. D. 2004. *An Introduction to Statistical Signal Processing*. Cambridge: Cambridge University Press.

Ito, K. 1951. On Stochastic Differential Equations. *Memoirs of the American Mathematical Society*, 4: 1–51.

Jhonson, D. H. 2013. *Statistical Signal Processing. Lecture Notes*. Houston: Rice University.

Karatzas, I. and Shreve. S. 1991. *Brownian Motion and Stochastic Calculus*. New York: Springer-Verlag.

Klebaner, F.C. 1998. *Introduction to Stochastic Calculus with Applications*. London: Imperial College Press.

Kloeden, P. E. and Platen, E. 1992. *Numerical Solution of Stochastic Differential Equations*. Berlin: Springer-Verlag.

Kubo, R. 1986. Brownian Motion and Nonequilibrium Statistical Mechanics. *Science*, 233(4761): 330–334.

Langevin, P. 1908. Sur la théorie de mouvement brownien. *Comptes Rendus de l'Académie des Sciences*, 146: 530. (English translation: Langevin, P. 1997. On the theory of Brownian motion. American Journal of Physics, 65: 1079.)

Maruyama, G. 1955. Continuous Markov Processes and Stochastic Equations,.*Rendiconti del Circolo Matematico di Palermo*, 4: 48–90.

Milstein, G. N. 1974. Approximate Integration of Stochastic Differential Equations. *Theory of Probability & Its Applications*, 19(3): 557–562.

Nigam, N. C. and Narayanan, S. 1994. *Applications of Random Vibrations*. Springer-Verlag.

Øksendal, B. 2003. *Stochastic Differential Equations: An Introduction with Applications*. Berlin: Springer-Verlag.

Papoulis, A. 1991. *Probability, Random variables and Stochastic processes*. 3rd Ed. New York: McGraw-Hill, Inc.

Rogers, L. C. G. and Williams, D. 2000. *Diffusions, Markov Processes, and Martingales: Volume 1, Foundations*. 2nd Ed. Cambridge: Cambridge University Press.

Roy, D. and Rao, G. V. 2012. *Elements of Structural Dynamics: A New Perspective*. John Wiley & Sons.

Roy, D. and Rao, G. V. 2017. *Stochastic Dynamics, Filtering and Optimization*. Cambridge: Cambridge University Press.

Simmons, G. F. and Krantz, S. G. 2006. *Differential Equations: Theory, Technique, and Practice*. McGraw-Hill.

Stratonovich, R. L. 1966. A new representation for stochastic integrals and equations. *SIAM Journal on Control and Optimization*, 4: 362–371.

Vanmarcke, E. H. 1983. *Random fields*. Cambridge, MA: MIT Press.

Vecer, J. 2011. *Stochastic Finance: A Numeraire Approach*. New York: CRC Press.

Wiener, N. 1923. Differential space. *Journal of Mathematical Physics*, 58: 131–174.

Wu, F. and Hu, S. 2008. Stochastic functional Kolmogorov-type population dynamics. *Journal of Mathematical Analysis and Applications*, 347: 534–549.

Appendix 5

A5.1 MATRIX $\mathfrak{g}$ AND THE CONNECTION MATRICES FOR THE RASTRIGIN FUNCTION IN EXAMPLE 5.2

The Rastrigin function is given in Equation (5.32) in Chapter 5. Let

$$f(\boldsymbol{x}) = \sum_{j=1}^{m} \left\{ x_j^{\,2} - 10\cos 2\pi x_j + 10 \right\} \tag{A5.1}$$

The first-order Euclidean derivative is:

$$\frac{\partial f}{\partial x_j} = 2x_j + 20\pi \cos 2\pi x_j \tag{A5.2}$$

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = 2 + 40\pi^2 \cos 2\pi x_j, \quad \text{if } i = j \text{ and zero otherwise} \tag{A5.3}$$

Now, the matrix $\mathfrak{g}$ associated with the Riemannian metric g is:

$$\mathfrak{g}_{ij} = 0.5 \frac{\partial^2 f}{\partial x_i \partial x_j} \tag{A5.4}$$

Matrices corresponding to the Cristoffel symbols:

The derivatives of the matrix $\mathfrak{g}$ are given by:

$$\frac{\partial \mathfrak{g}_{ij}}{\partial x_k} = 0.5 \frac{\partial^2 f}{\partial x_k \partial x_i \partial x_j}, \; k = 1, 2, \ldots, m \tag{A5.5}$$

where

$$\frac{\partial^2 f}{\partial x_k \partial x_i \partial x_j} = -80\pi^3 \cos 2\pi x_j, \quad \text{if } i = j = k \text{ and zero otherwise} \tag{A5.6}$$

The Christoffel symbols are obtained from the expressions:

$$\Gamma_{ij}^{k} = \frac{1}{2} \mathfrak{g}^{kl} \left(\frac{\partial \mathfrak{g}_{jl}}{\partial u^i} + \frac{\partial \mathfrak{g}_{il}}{\partial u^j} - \frac{\partial \mathfrak{g}_{ij}}{\partial u^l} \right), k = 1, 2, \ldots, m \tag{A5.7}$$

where $\mathfrak{g}^{kl}$ is the $(kl)^{th}$ element of $\mathfrak{g}^{-1}$.

A5.2 FIRST- AND SECOND-ORDER DERIVATIVES OF THE BUMP FUNCTION

Choosing the bump function as the kernel, one has:

$$\psi(\boldsymbol{y} - \boldsymbol{x}) = \exp\left(-\frac{1}{1 - (y - x)^T (y - x)} \right), x \in \mathbb{R}^n \tag{A5.8}$$

It follows that:

$$\frac{\partial \psi(|\boldsymbol{y}^j - \boldsymbol{x}|)}{\partial x_i} = -2\psi(|\boldsymbol{y}^j - \boldsymbol{x}|). \frac{(\boldsymbol{y}^j - \boldsymbol{x})}{\left(1 - (\boldsymbol{y}^j - \boldsymbol{x})^T (\boldsymbol{y}^j - \boldsymbol{x})\right)^2},$$
$$j = 1, 2, \ldots \tag{A5.9}$$

The derivative of the $\mathfrak{g}$ matrix is given by:

$$\frac{\partial \mathfrak{g}_{ij}}{\partial x_k} = \frac{\partial}{\partial x_k} \left(\frac{\partial E(\boldsymbol{x})}{\partial x_i} \right). \frac{\partial E(\boldsymbol{x})}{\partial x_j} + \frac{\partial E(\boldsymbol{x})}{\partial x_i} \frac{\partial}{\partial x_k} \left(\frac{\partial E(\boldsymbol{x})}{\partial x_j} \right) \tag{A5.10}$$

where

$$\frac{\partial}{\partial x_k} \left(\frac{\partial E(\boldsymbol{x})}{\partial x_i} \right) = \frac{1}{N} \left\{ \sum_j E(\boldsymbol{y}^j) \frac{\partial^2 \psi(|\boldsymbol{y}^j - \boldsymbol{x}|)}{\partial x_k \partial x_i} \right\} \tag{A5.11}$$

and

$$\frac{\partial^2 \psi\left(\left|\boldsymbol{y}^j-\boldsymbol{x}\right|\right)}{\partial x_k \partial x_i}=4\,\psi\left(\left|\boldsymbol{y}^j-\boldsymbol{x}\right|\right)\frac{\left(y_k^j-x_k\right)\left(y_i^j-x_i\right)}{\left(1-\left(\boldsymbol{y}^j-\boldsymbol{x}\right)^T\left(\boldsymbol{y}^j-\boldsymbol{x}\right)\right)^4}$$

$$+\,2\psi\left(\left|\boldsymbol{y}^j-\boldsymbol{x}\right|\right)\frac{1}{\left(1-\left(\boldsymbol{y}^j-\boldsymbol{x}\right)^T\left(\boldsymbol{y}^j-\boldsymbol{x}\right)\right)^2}$$

$$-8\psi\left(\left|\boldsymbol{y}^j-\boldsymbol{x}\right|\right)\frac{\left(y_k^j-x_k\right)\left(y_i^j-x_i\right)}{\left(1-\left(\boldsymbol{y}^j-\boldsymbol{x}\right)^T\left(\boldsymbol{y}^j-\boldsymbol{x}\right)\right)^3},\quad \text{if } \left(k\neq i\right) \tag{A5.12a}$$

and

$$\frac{\partial^2 \psi\left(\left|\boldsymbol{y}^j-\boldsymbol{x}\right|\right)}{\partial x_k \partial x_i}=4\,\psi\left(\left|\boldsymbol{y}^j-\boldsymbol{x}\right|\right)\frac{\left(y_k^j-x_k\right)\left(y_i^j-x_i\right)}{\left(1-\left(\boldsymbol{y}^j-\boldsymbol{x}\right)^T\left(\boldsymbol{y}^j-\boldsymbol{x}\right)\right)^4}$$

$$-8\psi\left(\left|\boldsymbol{y}^j-\boldsymbol{x}\right|\right)\frac{\left(y_k^j-x_k\right)\left(y_i^j-x_i\right)}{\left(1-\left(\boldsymbol{y}^j-\boldsymbol{x}\right)^T\left(\boldsymbol{y}^j-\boldsymbol{x}\right)\right)^3},\ \text{if } \left(k=i\right) \tag{A5.12b}$$

Knowing $\frac{\partial^2 \psi\left(\left|\boldsymbol{y}^j-\boldsymbol{x}\right|\right)}{\partial x_k \partial x_i}$ from the last equation, $\frac{\partial}{\partial x_k}\left(\frac{\partial E(\boldsymbol{x})}{\partial x_i}\right)$ is computable from Equation (A5.11) from which the derivative of $\mathfrak{g}$ matrix is obtained from Equation (A5.10).

A5.3 RIEMANNIAN GRADIENT OF LOG-LIKELIHOOD FUNCTION $l\left(\boldsymbol{\theta}_t;\boldsymbol{Z}\right)$ AND THE DERIVATIVES OF $\mathfrak{g}$ FOR THE EXAMPLE PROBLEM 5.4

The log-likelihood function $l\left(\boldsymbol{\theta}_t;\boldsymbol{Z}\right)$ is given in Equation (4.93) in Chapter 4. With $\mathbb{f}_{\boldsymbol{Z}}(\boldsymbol{z};\boldsymbol{\theta})$ given by Equation (5.42), the Euclidean gradient of the log-likelihood function $l\left(\boldsymbol{\theta}_t;\boldsymbol{Z}\right)t$ is given by:

$$\frac{\partial l}{\partial \boldsymbol{\theta}}\left(\theta_k; \boldsymbol{Z}\right) = \begin{pmatrix} \dfrac{n}{\alpha_k} + \sum_{i=1}^{n} \log\left(1 - e^{-\lambda_k z_i}\right) \\ \dfrac{n}{\lambda_k} - \sum_{i=1}^{n} z_i + \left(\alpha_k - 1\right) \sum_{i=1}^{n} \dfrac{z_i e^{-\lambda_k z_i}}{1 - e^{-\lambda_k z_i}} \end{pmatrix} \tag{A5.13}$$

The Riemannian gradient is given by $\mathfrak{g}^{-1} \frac{\partial l}{\partial \boldsymbol{\theta}}\left(\theta_k; \boldsymbol{Z}\right)$ where $\mathfrak{g}$ is the matrix associated with the Riemannian metric and it is the same as the Fisher information matrix $\boldsymbol{I}_n\left(\theta_k\right)$ for the estimation problem. See Equation (4.94) in Chapter 4 to obtain $\boldsymbol{I}_n\left(\theta_k\right) = E_{\boldsymbol{Z}}\left[\left(\frac{\partial l\left(\boldsymbol{\theta}; \boldsymbol{Z}\right)}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial l\left(\boldsymbol{\theta}; \boldsymbol{Z}\right)}{\partial \boldsymbol{\theta}}\right)^T\right]$ as:

$$\boldsymbol{I}_n\left(\boldsymbol{\theta}_k\right) = \begin{bmatrix} -\dfrac{n}{{\alpha_k}^2} & \sum_{i=1}^{n} \dfrac{z_i e^{-\lambda_k z_i}}{1 - e^{-\lambda_k z_i}} \\ \sum_{i=1}^{n} \dfrac{z_i e^{-\lambda_k z_i}}{1 - e^{-\lambda_k z_i}} & -\dfrac{n}{\lambda_k^2} - \left(\alpha_k - 1\right) \sum_{i=1}^{n} \dfrac{z_i^2 e^{-\lambda_k z_i}}{\left(1 - e^{-\lambda_k z_i}\right)^2} \end{bmatrix} \tag{A5.14}$$

With $\mathfrak{g} = \boldsymbol{I}_n\left(\boldsymbol{\theta}_k\right)$ and $\boldsymbol{\theta}_k = \left(\alpha_k,\ \lambda_k\right)^T$ at any k^{th} iteration, its derivatives with respect to the two parameters are:

$$\frac{\partial \mathfrak{g}}{\partial \alpha_k} = \begin{bmatrix} \dfrac{2n}{{\alpha_k}^3} & 0 \\ 0 & -\sum_{i=1}^{n} \dfrac{z_i^2 e^{-\lambda_k z_i}}{\left(1 - e^{-\lambda_k z_i}\right)^2} \end{bmatrix} \tag{A5.15a}$$

$$\frac{\partial \mathfrak{g}}{\partial \lambda_k} = \begin{bmatrix} 0 & \sum_{i=1}^{n} \dfrac{-z_i^2\left(1 - e^{-\lambda_k z_i}\right) e^{-\lambda_k z_i} - z_i^2 e^{-2\lambda_k z_i}}{\left(1 - e^{-\lambda_k z_i}\right)^2} \\ \sum_{i=1}^{n} \dfrac{z_i e^{-\lambda_k z_i}}{1 - e^{-\lambda_k z_i}} & -\dfrac{n}{\lambda_k^2} - \left(\alpha_k - 1\right) \sum_{i=1}^{n} \dfrac{z_i^2 e^{-\lambda_k z_i}}{\left(1 - e^{-\lambda_k z_i}\right)^2} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -\sum_{i=1}^{n} \frac{z_i^2 e^{-\lambda_k z_i}}{\left(1-e^{-\lambda_k z_i}\right)^2} \\ -\sum_{i=1}^{n} \frac{z_i^2 e^{-\lambda_k z_i}}{\left(1-e^{-\lambda_k z_i}\right)^2} & \frac{2n}{\lambda_k^3} - \left(\alpha_k - 1\right) \sum_{i=1}^{n} \frac{-z_i^3 e^{-\lambda_k z_i}\left(1-e^{-2\lambda_k z_i}\right)}{\left(1-e^{-\lambda_k z_i}\right)^4} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & \sum_{i=1}^{n} \frac{-z_i^2 e^{-\lambda_k z_i}}{\left(1-e^{-\lambda_k z_i}\right)^2} \\ \sum_{i=1}^{n} \frac{-z_i^2 e^{-\lambda_k z_i}}{\left(1-e^{-\lambda_k z_i}\right)^2} & \frac{2n}{\lambda_k^3} + \left(\alpha_k - 1\right) \sum_{i=1}^{n} \frac{z_i^3 e^{-\lambda_k z_i}\left(1-e^{-2\lambda_k z_i}\right)}{\left(1-e^{-\lambda_k z_i}\right)^4} \end{bmatrix} \quad \text{(A5.15b)}$$

Index